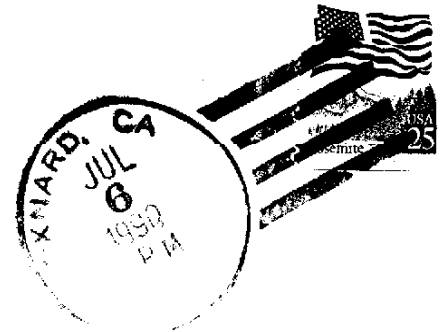


NEXT MEETING - JULY 9, 1990  
868 DECATUR, VENTURA, CA  
7:30 PM

**TICO TOPICS**

326 GLEN WAY  
FILLMORE, CA 93015



DALLAS TI HOME COMPUTER GROUP  
P.O. BOX 29863  
DALLAS, TX 75229

## EDITOR'S RAMBLINGS...

WELL IT HAD TO HAPPEN SOONER OR LATER. SO IF YOU WERE NOT AT THE LAST MEETING, PLEASE TAKE NOTICE. WE HAVE BEEN INFORMED THAT IF WE WOULD LIKE TO CONTINUE TO USE THE BANQUET ROOM AT THE RED BARON, WE WILL NOW HAVE TO PAY A RENTAL FEE. WITH THE SIZE OF OUR CLUB, WE WOULD SOON BE IN THE POOR HOUSE, SO IT WAS DECIDED THAT WE WILL HAVE TO MOVE. THE JULY MEETING HAS BEEN MOVED TO BOB MARTYN'S HOUSE. THE ADDRESS IS 868 DECATUR, VENTURA (SEE THE ENCLOSED MAP). THE DATE WAS ALSO CHANGED FROM JULY 4 (A HOLIDAY) TO JULY 9...PLEASE MAKE A NOTE OF THIS. WE WILL HAVE TO BE MAKING A DECISION AS TO WHAT PLACES WE WOULD LIKE TO GATHER IN. PERHAPS BY MEETING IN EACH OTHERS HOME THE NEED FOR A FULL SYSTEM WILL BE MET. SUMMERTIME IS AT HAND AND NOT EVERYONE WILL BE ABLE TO COME TO THE MEETINGS, BUT I STILL THINK THAT THE TIME WE HAVE IS WORTH THE EFFORT TO COME. HOPE TO SEE MANY OF YOU THERE.

GABRIEL ASENAS, EDITOR

## T I C O T O P I C S

THE OPINIONS EXPRESSED IN THIS NEWSLETTER ARE THOSE OF THE AUTHORS AND DO NOT NECESSARILY REFLECT THOSE OF THE OFFICERS OR MEMBERS OF TICO. PERMISSION IS GIVEN TO DUPLICATE PROVIDED THAT CREDIT IS GIVEN TO THAT AUTHOR AND TO THE PUBLICATION.

# AT THE APPLE'S CORE

MacFlix, a FULL review

by R. Coffey

The review of MacFlix last month was a quicky, and we will look into it much further with this review!

What is MacFlix? It is a program that will let us TIers manipulate MacPaint files. MacPaint files have a resolution of 576 pixels across by 720 down. (Yes, that picture of vehicles on the back of October Interface and the picture titled "Clip #7" this month is 576 by 720!) MacFlix let's you view the MacPaint files by windowing over them, much like Picasso and JoyPaint window over their own work area.

You can change the number of pixels your window moves over the MacPaint file, which makes it very easy to quickly get to the portion you are most interested in! You may then save that portion on the screen as a TI-Artist picture. (Or as a My-Art picture, if you have a Myarc 9640). Essentially, we are just saving a portion of the MacPaint file so we can use it on our own drawing programs (which is the main objective of this program, and the reason why I bought it!)

MacFlix will let you do much more than just that, though! It will print out the ENTIRE MacPaint file to your printer (Epson or Prowriter compatible). Epson compatables have 3 options for printout, where Prowriter gets only one. Option #1 (for Epson), also called Low Resolution, will produce a squat picture just like the single density output from Artist or Graphx. This option will let the far right 10% of the picture fall off the page. Option #3, High Resolution, spits out a very squished (horizontally) picture, but gives you the entire picture. The older printers, like Star Gemini 10-X, the original TI

printer, etc., will ONLY be able to use option #1 and #3. For those with a newer printer, like the Star NX-10, Star NX-1000, etc., you can use Option #2, Medium Resolution, which will give you the entire picture on the page, and is nearly perfectly proportional. (At the end, I'll tell you how to get PERFECTLY PROPORTIONAL print-outs!) Epson developed an additional set of new graphics after their early printers, and Medium Resolution uses this newer set.

So where can you get MacPaint files? Three files come on the Mac Flix disk, plus I got a bonus disk of 5 files (since I ordered it before October '88). I was able to download a few MacPaint files locally, but I will be going to the networks of Delphi and GEnie as a major resource of pictures. If you happen to have a friend with an IBM who has MacPaint files, they have included a conversion routine (for use with PC Transfer), so you can directly copy them from the IBM disk to your own. (PC Transfer requires CorComp or Myarc disk controller and double sided drives, also from Genial Computerware.) At the end of the manual, they list that the Boston Computer Society has many MacPaint files, but I have yet to explore that avenue. If demand (and supply) dictate, our club MAY start a MACPACK series (to complement our PICPACK series, but don't count on it just yet!).

MacFlix has a couple of other convenient features, like being able to Catalog a disk, which is very handy when you're not sure of the exact name of your file. You can also Delete files on a disk to clear room for an Artist picture you may be ready to save. Lastly, MacFlix will let you Invert the entire file that is currently loaded. I havn't used that last feature yet, but it could come in very handy!

Not all MacPaint files (or so I thought they were) are standard in size or format. MacPaint files normally have a few bytes at the beginning of the file that has a title of the picture, and is neccessary to load a MacPaint file in. Some files converted from the IBM may be lacking this title, and you can only load it into MacFlix after you have disabled MacFlix so it will not automatically look for this title. To do this, you need to press the function and (P) key together (in other words, the quote). Once you have pressed it,

the title is disabled for your entire session (until you reload the program again). Don't expect a flag on the screen to tell you have pressed it, because you get no video or audio feedback. If you can't load a file, you use this as your last resort!! And it can cause havoc if you're not loading a MacPaint file of some kind. A screen full of garbage comes to mind!

Many of you will be familiar with Archiver III by Barry Boone for our II, and other computers have their own compression programs. So you may find MacPaint files both on the Macintosh and IBM that are compressed in some way. These files will usually have a suffix to tell you this. These files are to be avoided, as you can't de-arc them with anything we have right now! When selecting files to download, keep your eyes open and you should be alright.

All MacPaint files are a fixed size (576 x 720) but MacPaint does a simple compression on them, so that the pictures that have any empty space in them will be smaller in size, depending on how much empty space there is. That is why you will see MacPaint files use different amounts of disk space, according to how sophisticated the graphics are in the picture.

With just an Extended Basic or Editor Assembler cartridge, MacFlix will handle a MacPaint file up to 24k in size. (They can theoretically be up to 51k, but 24k is enough space for most of them!) With a Myarc 9640, SuperCart, or GRAM Kracker (with the Write Protect switch off!) you can load a MacPaint file up to 32k in size.

If you have a file larger than available memory, then MacFlix will load in as much of the file as it can, and you will get garbage for the bottom portion of the file it was unable to load.

So, who can really use this program? Before even considering this program, you should check out all possible sources where you may be able to get MacPaint files from! Without a

source to get more MacPaint files from, it isn't worth your money, unless you like to pay for a nice program with only a handful of files to work with!

MacFlix offers to greatly expand our ability to get GREAT looking artwork from the biggest collection of artwork available. Genial Computerware has a winner here! For those who are interested in getting their hands on terrific artwork and are willing to hunt for the MacPaint files, it's well worth the price! The price is \$15, and can be had from Genial Computerware, P.O. Box 183, Grafton, MA 01519.

MACFLIX : A TECHIE TIP : The key to getting a perfectly proportional picture is to have the density of pixels the same both horizontally and vertically. A ratio of 1:1 to what you'd like, but not all printers offer it! In the vertical direction, printers (9-pin) have a density of 72 dots per inch. (So what you would like ideally is 72 per inch horizontally!) In Medium Resolution, Mac Flix selects the graphics mode of #4, which is 80 dots per inch horizontally. That is pretty good, but we can do better than that! Graphics mode #5 offers 72 dots per inch horizontally (exactly what we want!). In order to get MacFlix to choose #5 instead, you will need to go into a sector editor (Disk Utilities is one of the better ones) and search for the hex string "1B2A04". After you have found it, you'll want to change that "04" to a "05". Save that sector back out and you're all set! Now when you select Medium Resolution the MacPaint pictures will go across the full 8 inches of the page, like they were suppose to! Why J.P. Hoddie didn't do this in the first place is beyond me! Remember, only the newer printers have this capability!  
RJC-11/88

```

[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
[]
[] T.I. WRITER FILE KEEPING []
[]
[] BY: PAUL E. SCHEIDEMANTLE []
[]
[] P & A S O F T W A R E []
[]
[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[

```

To Tell you the truth Once Upon A Time my Writer Files were a TOTAL DISASTER! I couldn't find anything without a long drawn out process. Sometimes spending hours going through all those disks really was a hassle. And then one day I told myself this has got to STOP!

One of the first problems to overcome was FILE NAMES that were limited to 10 characters. So I decided that a menu program of some kind was in order. This way I could have file names that were as long as I wanted them to be. Also it would be great if all I had to do was change (2) letters of the MENU file name to load any file. So here is a simple method that I use for keeping my T.I. Writer files from getting out of hand.

The 1st file of each disk will be called OO\_WF\_00. By making a menu file we can then have file descriptions longer than 10 characters. Because We Read the menu instead of the disk directory. This is especially good if you have a TI WRITER LOADER other than the module.

We can save all the Menu files to a disk for quick searches.

Here are the easy steps to T.I. Writer File Keeping Mastery!!!

STEP 1  
-----

SETUP 2 DISKETTES AS FOLLOWS:

1. FILE DISK #1 = FOR MENU FILES ONLY. NAME THIS DISK..... "WTRFILE\_00". We will save a copy of our menu's to this disk for easy access. This way we only have to access

file is.

2. FILE DISK #2 = FIRST OF 99 FILE DISKS AVAILABLE. NAME THIS DISK....."WTRFILE\_01"

STEP 2  
-----

SETUP A MENU PROGRAM AS FOLLOWS:

1. NAME THIS FILE..."OO\_WF\_01"  
OO = MENU  
WF = WRITER FILE  
01 = FILE DISK #1
2. KEEP THE MENU FILE SIMPLE OR AS COMPLEX AS YOU WOULD LIKE IT!
3. EXAMPLE OF FILE:
  - A) LINE 1 = MENU FILE NAME:  
"OO\_WF\_01"
  - B) LINE 2 = BLANK LINE
  - C) LINE 3+4 = HEADER:  
FILE NO. DESCRIPTION  
-----
  - E) LINE 5-7 = YOUR DIRECTORY:
 

OO_WF_01	MENU
01_WF_01	CALL LOADS
02_WF_01	DISK INFO

THE ABOVE DESCRIPTIONS CAN BE AS LONG AS YOU WISH!!!!

4. SAVE TO DISK.

STEP 3  
-----

HINTS:  
-----

1. AFTER LOADING A MENU FILE:
  - A) YOU CAN SEARCH FOR A FILE BY USING THE 'Find String' function.
  - B) LOAD A FILE AFTER YOU FIND IT BY SIMPLY DOING THE FOLLOWING:

```

FCFN ?
LF <ENTER>
AND CHANGE THE
FIRST 2 CHAR'S
OF THE FILE
NAME DISPLAYED!

```

NOTE: BE SURE TO PUT THE RIGHT DISKETTE IN!!!!

2. GO FIND THOSE FILES!!!!!!!!!!!!!!

# DEBUGGING

BY

JIM PETERSON

When you have finished writing a program, the next thing you should do is to run it. And, very probably, it will crash!

Don't be discouraged. It happens to the very best of programmers, very often. So, the next thing to do is to debug it. And you are lucky that you are using a computer that helps you to debug better than some that cost ten times as much.

There are really three types of bugs. The first type will prevent the program from running at all - it will crash with an error message. The second type will allow the program to run, but will give the wrong results.

And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware, or with faulty hardware. As for instance, trying to run a Basic program, which uses character sets 15 and 16, in Extended Basic.

First, let's consider the first type. The smart little TI computer makes three separate checks to be sure your program is correct. First, when you key in a program line and hit the Enter key, it looks to see if there is anything it can't understand - such as a misspelled command or an unmatched quotation mark. If so, it will tell you so, most likely by SYNTAX ERROR, and refuse to accept the line.

Next, when you tell it to RUN the program, it first takes a quick look through the entire program, to find any combination of commands that it will not be able to perform. This is when it may crash with an error message telling you, for instance,

that you have a NEXT without a matching FOR, or vice versa.

And finally, while it is actually running and comes to something that it just can't do, it will crash and give you an error message - probably because a variable has been given a value that cannot be used, such as a CALL HCHAR(R,C,32) when R happens to equal 0.

The TI has a wide variety of error messages to tell you when you did something wrong, what you did wrong, and where you did it wrong. But, it can be fooled! For instance, try to enter this program line (note the missing quotation mark). 100 PRINT "Program must be saved in:"merge format."

And, sometimes you may be told that you have a STRING-NUMBER MISMATCH when there is no string involved, because the computer has tried to read a garbled statement as a string.

Also, the line number given in the error message is the line where the computer found it impossible to run the program; that line may actually be correct but the variables at that point may contain bad values due to an error in some previous line.

If the error occurs in a program line which consists of several statements, and you cannot spot the error, you may have to break the line into individual single-statement lines. This is the easiest way to do that - Be sure the line numbers are sequenced far enough apart. Bring the problem line to the screen, put a ! just before the first ::, and enter it. Bring it back to the screen with FCTN 8, retype the line number 1 higher, use FCTN 1 to delete the first statement and the ! and ::, put a ! before the first ::, and continue. Then, when you have solved the bug, just delete the ! from the original line and delete all the temporary lines.

Pages 212-215 of your Extended Basic manual list almost all the error codes, and almost all the causes of each one - it will pay you to consult these pages rather than guessing what

is wrong.

You may create some really bad bugs when you try to modify a program that was written by someone else - especially if you add any new variable names or CALLs to the program. Your new variable might be one that is already being used in the program for something else, perhaps in a subscripted array. I have noticed that programmers rarely use @ in a variable name, so I always tack it onto the end of any variable that I add to a program.

Also, the program that you are modifying may have ON ERROR routines, or a prescan, already built in. The ON ERROR routine was intended to take care of a different problem than the one you create, so it could lead you far astray - you had better delete that ON ERROR statement until you are through modifying.

The prescan had better be the subject of another lesson, but if the program has an odd-locking command !@P- up near the front somewhere, it has a prescan built in. And if so, if you add a new variable name or use a CALL that isn't in the program, you will get a SYNTAX ERROR even though there is no error. One way to solve this is to insert a line with !@P+ just before the problem line, and another with !@P- right after it.

When a program runs, even though it crashes or is stopped by FCTN 4 or a BREAK, the values assigned by the program to variables up to that point will remain in memory until you RUN again, or make a change to the program, or clear the memory with NEW. This can be very useful. For instance, if the program crashes with BAD VALUE IN 680, and you bring line 680 to the screen and find it reads CALL HCHAR(R,C,CH) just type PRINT R;C;CH and you will get the values of R, C and CH at the time of the crash. You will find that R is less than 1 or more than 24, or C is less than 1 or more than 32, or CH is out of range.

In Extended Basic, you can even enter and run a multi-statement line in immediate mode (that is, without a

line number), if no reference is made to a line number. So, you can dump the current contents of an array to the screen by FOR J=1 TO 100::PRINT A(J):: : NEXT J - or you can even open a disk file or a printer to dump it to.

You can also test a program by assigning a value to a variable from the immediate mode. If you BREAK a program, enter A=100 and then enter CON, the program will continue from where it stopped but A will have a value of 100.

You can temporarily stop a program at any time with FCTN 4, of course (the manual says SHIFT C, but it was written for the old 99/4), and restart it from that point with CON. Or you can insert a temporary line at any point, such as 971 BREAK if you want a break after line 970. Or, you can put a line at the beginning of the program listing the line numbers before which you want breaks to occur, such as 1 BREAK 960,970,980 Note that in this case the program breaks just BEFORE those listed line numbers. You can also use BREAK followed by one or more line numbers as a command in the immediate mode.

The problem with using BREAK and CON is that BREAK upsets your screen display format, resets redefined characters and colors to the default, and deletes sprites. So, it is sometimes better to trace the assignment of values to your variables by adding a temporary line to DISPLAY AT their values on some unused part of the screen. If you want to trace them through several statements, it will be better to GOSUB to a DISPLAY AT. And if you need to slow up the resulting display, just add a CALL KEY routine to the subroutine.

Sometimes, your program will appear to be not flowing through the sequence of lines you intended (perhaps because it dropped out of an IF statement to the next line!) and you will want to trace the line number flow. This can be done with TRACE, either as a command from the immediate mode or as a program statement,

which will cause each line number to print to the screen as it is executed. If used as a command, it will trace everything from the beginning of the program, so it is usually better to insert a temporary line with TRACE at the point where you really want to start. Once you have implemented TRACE, the only way to get rid of it is with UNTRACE.

TRACE has its limitations because it can't tell you what is going on within a multi-statement line, and it will certainly mess up any screen display. Sometimes it is better to insert temporary program lines to display line numbers. I use CALL TRACE( ) with the line number between the parentheses, and a subprogram after everything else 30000 SUB  
TRACE(X)::DISPLAY AT(24,1):X :: SUBEND

Some programmers use ON ERROR combined with CALL ERR as a debugging tool, but I can't tell you much about that because I have never used it. ON ERROR can give more trouble than help if not used very carefully, and I cannot see that CALL ERR gives any information not available by other means.

Sometimes you can debug a line by simply retyping it. It is only very rarely that the computer is actually interpreting a line differently than it appears on the screen, but retyping may result in correcting a typo error that you just could not see. In fact, most bugs turn out to be very simple errors.

When you are debugging a string-handling routine, don't take it for granted that a string is really as it appears on the screen - it may have invisible characters at one or both ends. Try PRINT LEN(M\$) to see if it contains more characters than are showing; or PRINT "\*"&M\$&"\*" to see if any blanks appear between the asterisks and the string.

There is no standard way to debug a program. Each problem presents a challenge to figure out what is going wrong, to devise a test to find out what is really happening.

Don't debug by experimenting, by

changing variable values just to see what will happen, etc. Even if you succeed, you will not have learned what was wrong so you will not have learned anything - and if your program contains lines that you didn't understand when you wrote them, you will have real problems if you ever try to modify the program. (Believe me, I speak from experience!)

The following announcement was received by Jonathon Livingston K. from 99-N-STUFF 'PUTERS, Ltd., and was downloaded from the TIBBS NORTHWEST of Seattle, Washington.

Phillip Noah Blarney, Pres.

Dear TI-99 User,  
Ever since Texas Instruments thought it best to discontinue their fine product and leave us out in the cold to fend for ourselves, people like you and me have been asking for an upgrade to our miserable little machines. Well sir, your waiting is over. Announcing the new upgrade for your TI-99, it's the TI-99S6 (\$6 for Super Good!) It is, in our opinion, the best little buy that can be legally had in this country. You see, friend, my chief engineer and next door neighbor, Farley, had his kid's TI open trying to clean the cat fur from between the keys, and he looked up at me and said, "Phil, this ain't so hard to clone. Why, give us a little time and we can make one of them upgrades ourselves. If we could get it working before the next presidential election, we ought have it before them boys back east ever finish with theirs."

"You, know, Farley," I replied, "you got something there." So Farley and me, we cleaned out my garage and began working on what would become the TI-99S6! But enough preachin, let's get down to brass tacks.

The 'puter is about as big as my cat Fred, so if you ever met Fred you probably have a good idea. Either way it's about the size of a toaster oven, with a long air conditioner cord which hooks to 220 so you can plug it in next to your dryer. And unlike the 99/4, we put a fan in ours to keep her cool. M:



got the fans on sale too. Ever wonder what they do with those WWII office fans that's as big as a plate and painted that dirty grey? We'll, they're just the thing to keep all this high tech stuff cool.

Let me tell you, the cabinet is a beaut.

We went out and bought a lot of that cork board they have on sale over at the lumberyard, and we fashioned us this box. We then stuck some contact paper over it, shined her up with some wax and a little spit, and I'll eat spoiled milk if it doesn't look as good as a mantle clock polished with bacon fat. We also went down to the auto body shop and got us one of them smoked glass things all the VW's seem to have, and with a lick of the torch we fashioned a custom dust cover. Course, you have to prop her up with a pencil and a piece of postal tape, but Farley says we can throw that in without upping the price any. For those of you technical folks, well, we got a surprise. Farley got one of them old microwave ovens and he put together our microprocessor, the CPU5204-XOIR/PS199x. It's about the size of a pack of cigarettes and has more wires coming out of it than a beagle has hairs on his butt. We figure that you can just hook whatever you want to whatever wire and get better results than all that Japanese dip switching stuff.

Now, a lot of jaw flappin's been going on about keyboards, and we came up with a great idea. Farley said we have to keep costs down so we decided that we'd fix it so you can get one of those cheap typewriters at a rumage sale and hook it up straight away. You take the keys and you take that long little arm, and instead of hooking it to those levers, you hook it up to the little arms we have jutting out of the box, and you can use your typewriter as a keyboard. You can even use one of them electric jobs if that's your fancy. You just open up the bottom, pull off all that electrical junk and hook her up just like it was a manual job and you're all set. We also have a RAT (Really Astonishing Thingamajig) which'll beat those "mice" paws down. Instead of having its tail going all over the floor and hooking up to the 'puter itself, we decided to go remote control. Now what you do is you lift these two little antennas which are where its ears ought to be, and then you

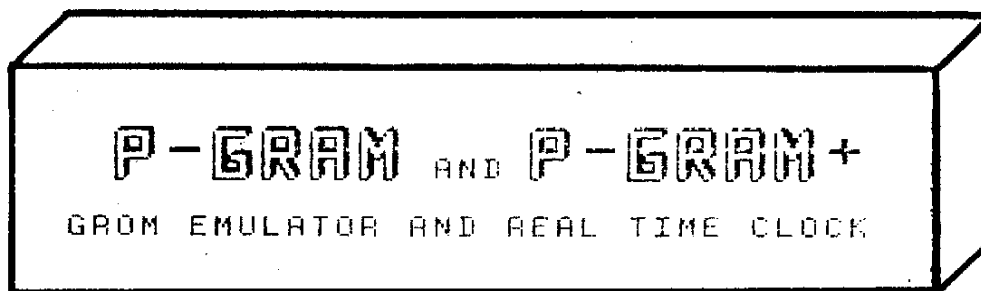
kind of give it thump on the back of its neck with the back end of your middle finger. The RAT will send a signal to the 'puter, which will send a signal back. The RAT's eyes'll light up like a Christmas tree and you're ready for business. Whole thing doesn't take more than thirty seconds or so to do. Of course, there's always someone yammering for compatibility. Our computer will become compatible with whatever is out there. Of course, it'll take time, and Farley figures that by the time we get it to run right our competition will be filing chapter 11. So we'll burn that bridge when we come to it.

Now we have to admit that the computer isn't quite up to snuff yet. We've found that if the computer is left on for more than five minutes it starts a minor electrical fire, and the CPU does have a tendency to explode, which sends the RAT running in circles underneath the house. However, we feel these are negligible bugs and will be worked out by the time you read this. In the mean time though, we've been traveling about showing off our dandy cabinet and telling good folks like yourself all about it and what it'll do when we get it to working proper. And I must admit, everyone seems pleased as pie with the cabinet and say if the 'puter is as good as the cabinet looks, we'll be living in Rio this time next year. And I thank you all for your whole hearted support. Maybe we can drop by your user group meeting and show you our cabinet too. We also have lots of literature, such as you're reading right now, and we have a few photos of Farley and me holding up the RAT. Those always seem to do well. Hope to see your check for the new 9936 computer soon. And God bless.

Cordially,

Phillip Noah Blarney

P.S. One of our admirer's suggested we include a garbage disposal in the beautiful cabinet as an option. Farley was so disappointed by this oversight on his part that to make amends he will not only include a garbage disposal but also a PAPER SHREDDER! This should be especially attractive to newsletter editors. We listen. Give us your feedback.



Designed by John Guion and Robert Jones / Produced by Horizon Computer / Distributed by Bud Mills Services

#### WHAT IS A P-GRAM? and WHAT IS A P-GRAM+?

The P-GRAM is a card for the Peripheral Expansion System that adds 72K of battery-backed memory to the TI-99/4A (40K GRAM, 16K bank switched module RAM, and 16K bank switched DSR RAM). This memory is added in place of the ROM memory used for module software. The P-GRAM allows you to save modules to disk and then load them into the P-GRAM's memory to be used. Once a module has been saved to disk and loaded into the P-GRAM, it won't be needed again. The computer cannot tell the difference between a module loaded into the P-GRAM and one inserted into the computer. Since the P-GRAM's memory is maintained by battery, the contents of the P-GRAM will remain even if the computer is turned off. THE P-GRAM+ ADDS 120k of GRAM to the P-GRAM to enable the VOLUME MODULE LIBRARY and ADDS three pages of TI Title Screens.

The real-time clock option may be purchased at an extra cost and provides the computer with time, date, and day-of-the-week information. The optional clock is compatible with software written for EITHER the MBP clock card or CorComps Triple Tech and 9900 Stand Alone clocks, thus providing compatibility with a wide range of existing clock-based software.

#### WHAT CAN THE P-GRAM DO FOR ME?

The P-GRAM may be used to emulate almost any module (including Extended BASIC, Editor/Assembler, Multiplan, TI-Writer, and hundreds of others!). This not only provides a backup of each module you currently own, but puts an end to frustrating problems caused by "flaky" modules and worn module ports.

The software required to save and load modules is part of the P-GRAM's operating system and is loaded when the card is installed. A few keystrokes are all that are needed to call up a menu-driven program that allows you to use any P-GRAM feature.

The files created when saving modules to disk for use with the P-GRAM are compatible with files saved by the Graa Kracker and Cart Saver programs. Thus, modified modules used with other devices can be used on the P-GRAM.

The P-GRAM+ has three additional pages of GRAM (five 8k banks per page) that allow you to store many GROM based modules in addition to the 72k P-GRAM capacity. You can also use J.P. Modie's "GRAM PACKER" to convert many of your favorite programs into GPL format and "stack" them into the P-GRAM+ for "instant access" from the TI Title screen. John Johnson's BOOT ver 12 will run from the P-GRAM+ as well as from any other device.

Since the P-GRAM uses RAM and GRAM memory to store modules, it can also be used to modify them using an advanced memory editor that is part of the P-GRAM's operating system. This allows bugs to be fixed (such as printer and RAM-DISK incompatibilities) and new features can be added or software can be customized to fit your specific needs. YOU no longer have to be satisfied with whatever was programmed into the original module. The P-GRAM allows you to change and improve things that you never could before.

The P-GRAM can be used just like a "Super Cart" module (an Editor/Assembler module with 8K of RAM). This allows the user to run the growing number of programs requiring this type of module. However, the P-GRAM's memory is not limited to running modules. The memory can be used for an application requiring RAM or GRAM memory. A full 56k of memory is available for use in the module memory space.

The built-in memory editor allows you to inspect or modify any memory accessible by the computer. Although the memory editor is designed primarily for making changes to modules, it can be used to access memory in any part of the system. The memory editor allows you to view, alter, move, fill, print, search, or dump to disk any memory you wish and also provides control over the CPU interface. The memory editor is simple to use, even for people who have no experience with such programs. Each function is documented and easily accessed using function keys.

The optional real-time clock allows your computer to easily access time and date information for use by a variety of programs. Since the P-GRAM's clock is compatible with both the MBP and CorComp clock devices you can use many existing programs requiring a clock device as well as create new ones. The P-GRAM clock is easy to access through any programming language and has built-in software to set the clock.

#### DO I HAVE TO BE AN "EXPERT" TO USE THE P-GRAM?

The P-GRAM is designed to be easy for anyone to use, regardless of prior computing experience. All software is menu-driven and user-friendly. Since the P-GRAM is completely software-controlled and uses no switches, the user only needs to follow the simple prompts to use any P-GRAM feature. The detailed operating manual describes step-by-step procedures for installing and using the P-GRAM and explains how to use every function of the P-GRAM card.

Of course, the P-GRAM is not limited by its ease of use. The operating manual includes an extensive technical data section with complete information on accessing the P-GRAM's features through BASIC or assembly language and includes sample source listings. The method of operation and control of the card is discussed in detail along with helpful advice for writing custom utilities. Additionally, complete source code for the operation system and its loader (including the memory editor) are provided on disk with the P-GRAM.

#### HOW CAN I GET A P-GRAM?

The P-GRAM is available with or without the real-time clock and may be purchased in kit form or fully assembled. The kit comes complete with circuit board, all parts, and an illustrated instruction guide. Assembled cards are fully tested and include a 6-month warranty. Kits and completed cards may be ordered from:

Bud Mills Services  
166 Dartmouth Dr.  
Toledo, OH 43614

Complete P-GRAM (72k) kit = \$150.00 Fully assembled = \$180.00 ADD \$20.00 for the CLOCK option.

Complete P-GRAM+ 192k kit = \$250.00 Fully assembled = \$280.00 ADD \$20.00 for the CLOCK option.

The P-GRAM requires a TI-99/4A, Peripheral Expansion System (P-Box), 32K, Disk Drive and Editor/Assembler (used for loading the operating system).

5% discount on orders of five (5) units or more. OHIO RESIDENTS add 6% sales tax. U.S. and Canada shipping included in price. Other foreign orders add \$3.00 surface or \$12.00 airmail.

Visa, MC, or American Express add 10%. Call 1-800-776-4951 (Disk Only Software) for credit card orders. Credit card orders may also be called in to BUD MILLS SERVICES on (419) 385-6946, we still add the 10% cc surcharge.

---

**IMPORTANT:** The P-GRAM does not currently function with QI (Quality Improved) consoles. If you have a tan-colored console, look into the I/O port on the right side of computer. If the connector is surrounded by silver "fingers", the console is a QI unit and will not function properly with the P-GRAM. If the connector is surrounded by gold "fingers", it is not a QI unit and is compatible with the P-GRAM. Black and silver consoles are compatible with the P-GRAM.