

Vol. 10 No. 5
May 1987

THE MSP 99 NEWSLETTER

Multiplan Printer Formatter

MPCTRL is a file you can load in with MULTIPLAN that allows you to make use of the fonts and control functions of your printer when outputting your results for reports or charts.

You start by loading MPCTRL into MULTIPLAN. The control codes are located in the upper left hand corner of the spreadsheet.

!!! IMPORTANT !!! At this point save the file to disk under the name you will be calling this MULTIPLAN FILE. [This way you will not lose the MPCTRL file].

You then move the codes from their current location, with the MOVE command, down to the bottom, or off to the right, or somewhere out of the way of the area you will be using [C200-R1, C1-R55, etc.]. You go ahead and do your spreadsheet like you want it and copy in to the cells where you want to with the commands that you wish to send to your printer [italics, condensed, expanded, etc.]. You do this with the COPY command so that if you wish to print another report with another type of format you can just copy the command for your printer from the cell it is in to the appropriate location in your spreadsheet.

When you go to print your report from your spreadsheet, just be sure to print only that part that you are using and not that part where the printer commands are located.

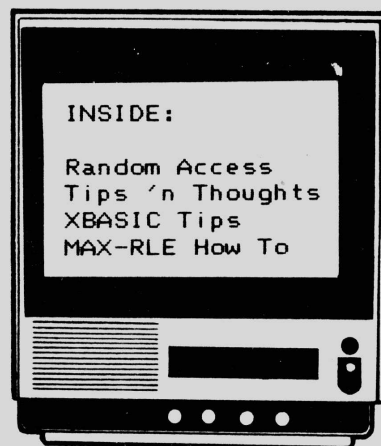
I hope this information is sufficient for your usage of the MPCTRL file.

MPCTRL is available in the MSP 99 Users Group Library.

RAFFLE UPDATE

Lots of interesting things happened at the April meeting and unfortunately not all of them good, but the monthly raffle went off without a hitch. Bob Culver went home that night with a copy of Donn Granros' "Old Dark Caves" tucked under his arm and I know he'll have lots of fun with it. (By the way, if you were hoping to win that prize, you haven't missed out completely. We'll be raffling another copy at the June meeting.)

This month we have something very special to offer. One of our members who shall remain nameless, (Herman Saul) has kindly donated a TI99/4A Home Computer to be raffled off at the May meeting. Imagine, a TI computer with all the cables for only \$1.00. I know that the price has been dropping on these since it was orphaned but ... Anyway, you must be at the meeting to win since this is the only place you can purchase the raffle tickets. I guess this means we'll be seeing all of you at the next meeting.



The MSP 99 USERS GROUP meets each month for discussions and presentations that enable its members to be better informed about their computers. Users group members share and exchange information. Some members have a broad range of computer expertise, others are just beginning. We are not affiliated with or sponsored by any other group or company. Membership dues are \$18 a year for a family or individual, and \$50 for a sponsor member. You're welcome to visit a meeting as a guest before you join. Call or write for more information.

USERS GROUP MEETINGS are held the third tuesday of each month at Dunwoody Industrial Institute, 818 Wayzata Blvd., Minneapolis, MN 55403. Meetings start at 7:00 PM.

MSP 99 USERS GROUP
P.O. BOX 12351
ST. PAUL, MN 55112, U.S.A.

PRESIDENT: Dick Lauhead 429-5256
V. PRES: George Madline 341-3780
SECRETARY: Gary Gese 529-3989
TREASURER: Mark Tellevik 780-5856

The MSP 99 NEWSLETTER is published eleven times per year on a monthly basis, except during July, by the MSP 99 Users Group. Members are encouraged to contribute articles for publication. Opinions expressed are those of the writers and not necessarily those of the MSP 99 Users Group, its officers, editors, or members. Materials accepted by the editors for publication in the MSP 99 Newsletter, including software listings, are believed to be in the public domain. Newsletter articles may be reproduced by other users groups if appropriate credit is given to the author (if one is listed), and to the Minneapolis, St. Paul 99 Users Group.

NEWSLETTER EDITOR

Gary Gese 529-3989

Articles intended for the next newsletter should be submitted NO LATER than the users group meeting on the month prior to publication. Articles submitted after this deadline are likely to appear in the following month's newsletter.

COMMITTEE VOLUNTEERS are sought for all of our committees. (Education, Equipment, Program, Publicity, Software, Newsletter) If you would like to join one of these committees or have an idea for a monthly program, please contact one of the officers.

COMMERCIAL ADVERTISEMENT RATES: Business firms that wish to communicate with our members may do so by placing an advertisement in the newsletter. Rates are: Full page \$40; Half page \$30; Quarter page \$22.

Each ad must be camera ready in one of the sizes indicated and paid in advance. Inserts (printed by the advertiser on 8 1/2 X 11 or 8 X 10) may be inserted in the newsletter at \$20 per sheet. Contact the editor for more information.

CHANGE OF ADDRESS: Before you move, please mail a change of address to the Users Group. DO NOT rely on the standard Post Office change of address card since the P.O. will not forward this Newsletter.



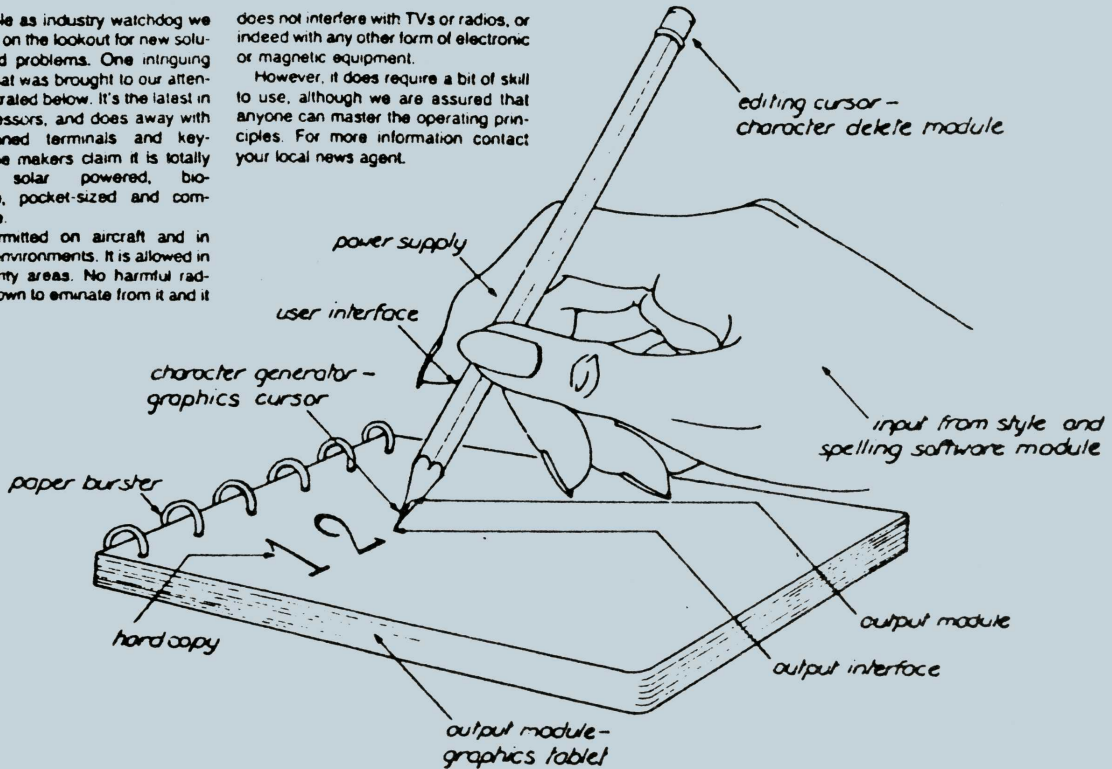
The latest in Word Processors.

In our role as industry watchdog we are always on the lookout for new solutions to old problems. One intriguing example that was brought to our attention is illustrated below. It's the latest in word processors, and does away with old fashioned terminals and keyboards. The makers claim it is totally portable, solar powered, biodegradable, pocket-sized and completely safe.

It is permitted on aircraft and in explosive environments. It is allowed in most security areas. No harmful radiation is known to emanate from it and it

does not interfere with TVs or radios, or indeed with any other form of electronic or magnetic equipment.

However, it does require a bit of skill to use, although we are assured that anyone can master the operating principles. For more information contact your local news agent.




MSP 99 Calendar of Events


MAY 19: XB II -- Tonight we get a peek at the new Extended BASIC II from Myarc. What it can do and how is it better than what we're used to. Find out tonight. We'll also be raffling off a TI 99/4A Computer.

JUNE 16: FORTRAN -- A NEW (?) programming language finally available for the TI 99/4A Home Computer. Come on down and find out what it's all about.

JULY 21: MYARC 9640 GENEVE -- That's right! If all goes as planned we should be able to get a good close look at one of these little goodies tonight. (Murphy permitting.) You won't want to miss this one!

Subgroup Meetings

ASSEMBLY GROUP -- 1st Tuesday of month, 7:00 p.m.
Bryant Community Center
Bryant Ave and 31st St.

BUSINESS and APPLICATION SIG
Call Dick Clemetson (926-8083)

EDUCATION -- At monthly meetings

YOUTH GROUP -- At monthly meetings

Committee Chairs

EQUIPMENT -- George Madline
(784-2395)

NEWSLETTER -- Gary Gese
(529-3989)

PUBLICITY -- Dave Wunderlin
(544-8266)

SOFTWARE -- Steve Gonnella
(533-8494)
6281 Winnetka Ave
Brooklyn Park, MN 55428

YOUTH GROUP --
Ed Johnson (690-3442)
Gordy Myers (377-6713)

FOR SALE BY MSP 99 GROUP

Keyboard ID strips for various TI programs.

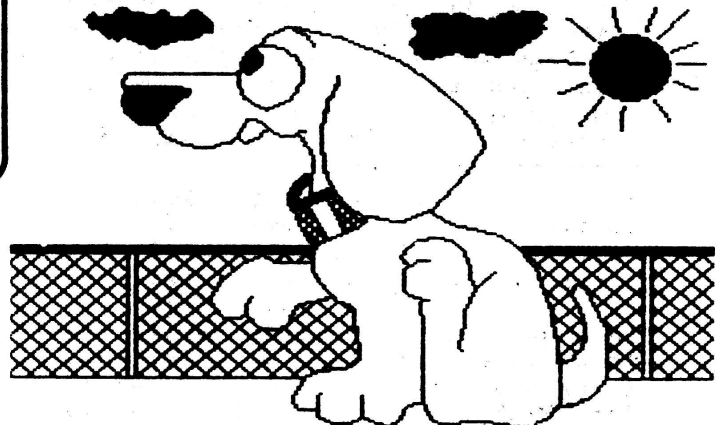
Double sided strips are .75 each:

- TE II and MULTIPLAN
- TE 3 DUMB TERMINAL and TE 3 ADM3A TERMINAL
- EDITOR/ASSEMBLER/BASIC and FORTH

Single sided strips are .50 each:

- p-SYSTEM
- EDITOR/ASSEMBLER/BASIC (this strip is adhesive backed to avoid loss)
- CONTROL DATA PLATO ACCESS DISK

Custom strips made on special order for \$1.00 per strip. Contact any Club officer or Tom Fairbairn. If several people request strips for a specific program that we do not stock, a quantity can be made to sell at the regular price.



RANDOM ACCESS

By Dick Lauhead

One of the simplest hardware projects for our computer is the installation of a load interrupt switch. I imagine thousands of speech synthesizers and consoles have been modified to incorporate this switch. Once installed, however, most users realize they don't have the foggiest idea how to use this new "feature". Hopefully, this article will help you understand what a load interrupt is, and how to make better use of the load interrupt switch. You will better understand the following discussion if you have at least some knowledge of assembly language, but you might find the information interesting even if you don't fully understand it.

One of the input pins on the 9900 microprocessor chip in the console is labeled NOT LOAD, which I'll call -LOAD. The NOT means the signal is active if low. Since ground is low, every time you press the load interrupt switch (which grounds this pin), you activate this signal. If -LOAD is low when the microprocessor has just completed executing an instruction, the load interrupt will be executed instead of the next program instruction.

So, what does the microprocessor do when the -LOAD pin is low? You may find it surprising, but it executes only ONE instruction! That instruction is BLWP @>FFFC. That's it--the rest is up to the programmer. Lest you fear that all the work of installing the load interrupt switch was for naught, let me hasten to point out that BLWP is an extremely powerful instruction. BLWP is the assembly language mnemonic for branch and load workspace pointer. This instruction is similar to mainframe instructions used to process many programs concurrently. It allows you to branch to another program, complete with its own set of registers, AND to get back to the original program where it left off.

In order to understand what the load interrupt can do for you, therefore, you have to understand the BLWP instruction. BLWP causes what TI calls a context change.

This term is not very descriptive. Control Data calls their somewhat similar instruction an exchange jump. This is a bit more descriptive of what happens when a BLWP is executed. The data at addresses >FFFC and >FFFE (in the case of a load interrupt) are exchanged for what was in the workspace pointer register (WP) and program counter register (PC). The old values that were in WP and PC are put into the new R13 and R14. Also the status register is stored in the new R15. This is all the information that is necessary to return to the interrupted program. By "new" I mean the registers in the new workspace that was loaded from the data at >FFFC.

The chart that follows shows what happens if the load interrupt button is pressed while an instruction is being executed at address >A120 with workspace at >A000, assuming the instruction is only 2 bytes long and is not a branch or jump. Note that PC usually points to the NEXT instruction to be executed.

	BEFORE:		AFTER:
WP	>A000----->	R13	>A000
PC	>A122----->	R14	>A122
STATUS	>XXXX----->	R15	>XXXX
>FFFC	>8300----->	WP	>8300
>FFFE	>602C----->	PC	>602C
		>FFFC	>8300
		>FFFE	>602C

Before the button is pressed the instruction at >A120 is being executed. If the button were not pressed, the instruction at address >A122 would be executed next. However, the load interrupt in this case causes the program to continue at address >602C with a new workspace at >8300. Addresses >FFFC and >FFFE must, of course, have been loaded before the interrupt occurred. Those address can be set with a debugger or from a running program.

The above example is a real one. I use it quite often to jump into Superbug II in my super cartridge for debugging assembly programs. The data in addresses >FFFC and >FFFE can be set to anything you wish. They are often set to point to a screen dump program in memory, but can be used any way you like. As an experiment, set >FFFC to >8300 and >FFFE to >A000 with a debugger. Now load FAST-TERM or any other program that starts at

address >A000. Press QUIT to return to the color bar screen. Now press the load interrupt button. You should immediately see the FAST-TERM initial display. Whatever address is in address >FFFE when the button is pressed will be executed immediately. The load interrupt is a non-maskable interrupt. That is, it cannot be disabled even by a LIM1 0 instruction. Only RESET has a higher priority than -LOAD. If RESET and -LOAD signals occur simultaneously, RESET will happen first, and then the load interrupt will be processed.

With the simple load interrupt switches most of us have installed, it is not possible to return to the interrupted program without using special software techniques. The switch bounce causes multiple interrupts which corrupt R13, R14, and R15. Remember, the return information was stored in those 3 registers by the BLWP instruction. In many cases you will not care to return anyway, so this would not be a problem. If you have a load interrupt switch that produces only one interrupt, you can return to the original program by simply executing a RTWP (return with pointer) instruction, assuming you have not reloaded the workspace pointer register. RTWP takes the value in R13 and places it in the WP register, R14 and places it in the PC register, and R15 and places it in the STATUS register. This has the effect of restoring the environment present when the load interrupt occurred. The interrupted program will then resume execution where it left off.

The load interrupt switch circuit to produce only one load interrupt is quite simple. It consists of 2 chips, 3 resistors, a diode and a different pushbutton switch. If there is enough interest in this circuit, we could publish it in a future newsletter. By the way, don't waste your time building the load interrupt circuit described in the March issue of Micropendium (page 12). It simply gives precisely timed switch bounces. It does NOT give you a one shot load interrupt! Meanwhile, if you come up with any interesting load interrupt software, I would very much appreciate receiving a copy of it. Very little has been done in this area. You could become famous!

RED FACE DEPT.--Although the Easter calculation program in the last newsletter was not intended as an April fool's joke, it certainly made a fool out of me! If you read the article carefully, you will see that I stated that Easter can occur no later than April 25. You will also note that the example year (1992) shows Easter as April 26. There are several algorithms for calculating Easter, but obviously this one belongs in the trash can. The following program works for all years (I hope!). I tested it for the years 1583-2099 and it produces no dates outside the legal range. (The parentheses around numbers following an asterisk can be omitted. They are there only to get around a "feature" of the TI Writer formatter.)

```

100 CALL CLEAR
110 INPUT "ENTER YEAR: ":Y
120 IF Y<1583 THEN PRINT "YEAR MUST
BE >1582": :GOTO 110
130 Y1=Y/19
140 A=INT((Y1-INT(Y1))*(19)+.001)
150 B1=Y/100::B=INT(B1)
160 C=INT((B1-INT(B1))*(100)+.001)
170 D1=B/4::D=INT(D1)
180 E=INT((D1-INT(D1))*(4)+.001)
190 F=INT(((B+8)/25)+.001)
200 G=INT((B-F+1)/3)
210 H1=(19*A+B-D-G+15)/30
220 H=INT((H1-INT(H1))*(30)+.001)
230 C1=C/4::I=INT(C1)
240 K=INT((C1-I)*(4)+.001)
250 L1=(32+2*E+2*I-H-K)/7
260 L=INT((L1-INT(L1))*(7)+.001)
270 M=INT((A+11*H+22*L)/451)
280 N1=(H+L-7*M+114)/31::N=INT(N1)
290 P=INT((N1-N)*(31)+.001)
300 N$="APRIL"
310 IF N=3 THEN N$="MARCH"
320 PRINT "EASTER IS ";N$;P+1: :
::PRINT "WANT ANOTHER YEAR? (Y/N)"
330 CALL KEY(3,K,S)::IF S=0 THEN
330 ELSE IF K=89 THEN 100

```

E
A
S
T
E
R

D
A
Y



TIPS 'N' THOUGHTS

by Tom Fairbairn

I mentioned in another column that I would like to continue with the discussion concerning the TI-Writer Formatter. I don't intend for this column to become devoted just to the TI-Writer program, but there is so much involved with it that needs to be viewed that it may take two or three months to cover it all.

This month I would like to go into a feature of TI-Writer that will allow the creation of book-type documents that are too long for the memory in the TI to handle. While the memory space in our systems, even with expansions, is severely limited, the writers of this program realized the problem and made provisions to get around it.

To use the capability, you have to use the features of the Formatter. Once again, the ruler documents can be of real help.

Generally, we put the ruler document at the beginning of the work we are entering. This is fine if the production document will fit entirely into the TI's storage. But if your production document grows to be more than about eight standard pages, you will get the infamous "BOOOOP" and a nastygram that boils down to, "You are using too much storage, partner!"

Generally, when you start on a document you have a pretty good idea if it is going to be a one-page business letter or rival the Encyclopedia Britanica. If you know it's going to be a long one, you might want to choose a name that is sized so you can stick numbers out behind the name. In this case, you would normally use a ruler that has only the file segment name, date, and version as comments, and the usual tab/margin ruler. The first segment of the file group will be 1, the next 2, etc. The overall file will be named without the numbers later.

Limit the number of pages you stick into a file segment. In most cases, if you are entering 65-character lines (margins at 10 and 75, which is common), and using 57-line pages at 6 lines per inch (also common, as it gives proper

bottom margins), you will end up able to handle just about 8 pages per segment. If you changed any of the given values, of course, this will change also.

Still using the parameters above, when you get close to 500 lines of text and Formatter commands, you better think about ending the current file and starting another.

Once you have entered all your segments, you want to print them out as a consolidated volume. Here is where the Formatter comes into play.

You need to create a ruler document that sets the rules for the entire printout. To do this, read in the conventional ruler document you would normally use for control of a unified document. The text is going to look a bit strange, though.

If the overall name of the file you are building is MYBOOK, then you will likely have called your segments MYBOOK1, MYBOOK2, MYBOOK3, and so on. The ruler document is then called MYBOOK when you go to save it.

The test is entered in the manner given next.

```
.IF DSK2.MYBOOK1
.IF DSK2.MYBOOK2
.IF DSK2.MYBOOK3
```

All Formatter commands given in the ruler document, including the transliterations, will be in force while the segments, or subordinate documents, are printing. In addition, if you have defined headers and footers in the ruler document, they will appear on all pages of the printout, with all page numbers consecutive just as if we had one continuous print file.

If any Formatter commands appear in the subordinate documents, they will be recognized ONLY IF they are not defined in the ruler document.

You can set controls in the subordinate files, but be sure they are not trying to argue with the ruler or they will be ignored.

You may have noticed that I set up all calls from a single document. There is a temptation to call document 2 from 1, and 3 from 2, and so on, chaining the calls from

each document to the next. Don't do it, it won't work. The program can't handle chaining. It wants just one ruler, and that is all it will allow!!

There are other variations of this process that are explained on pages 109 and 110 of the TI-Writer manual, but this will get you started on writing the novel of the century if that is what you want to do.

Another aspect of printing a manual or book is the use of headers and footers. Headers are the more-or-less fixed information printed at the top of every page in the document. Footers are similar material that is printed at the bottom of every page.

There are two basic types of headers, called "running" and "section" or "chapter" headers. Running headers appear on every page of the book or document. Section or chapter headers change every few pages as the subject matter changes. Once in a while you can see a combination of both, usually on two lines, where the top line is the running header with the book title, and the lower one is the section header. In periodicals such as newspapers and magazines, the page numbers usually appear in the header. In books, the page numbers can either be in the headers or the footers. The most common practice is to put the page numbers at the bottom in books. It is also common to have the page numbers and/or headers always print at the open (rather than the bound) edge of the page. When the header or footer information changes sides of the page, it is usually referred to as "alternating" headers or footers.

Some systems (the one I use at work is an example) have formatters that are designed to do alternating headers and footers on demand. However, TI-Writer had to be able to fit into the limited memory space of the machine, and this is one of the little niceties that had to give way to memory space.

Not to be discouraged; with some effort you can still accomplish the printing of alternating headers and footers with a bit of planning and work. They look every bit as good as the professional ones. This can be accomplished in either of two ways.

We have discussed ruler documents at length. Well, if you remember, the ruler documents can be called into a production document at any time. You can set up two ruler documents that contain the alternate headers, and indications for when you have reached the end of the page so you leave room for a footer that will be supplied by the master printing ruler document. As you start each page, read in the proper ruler document for the page you are working with (even or odd page number, odd pages on the right of the binding). If you do this, it is best to use the .BP command a couple of lines above the absolute last line so you can add a few words to a page without having to redo the whole section.

The other approach is to use the Mailmerge feature of TI-Writer. The use of this feature is explained in the TI-Writer manual beginning on page 111 and going through page 113. I build a value file that contains the elements I will use for the entire book (running heads, section heads, and so forth). Then I go enter my book, using however many sections are needed. After the book is entered, I go back through it and count the printable lines, putting in page breaks (the .BP command to the formatter) when the line count reaches the number of lines I can print on a page. This leaves room for the headers and footers on each page. Then I go to the top of each page, and position the alternate input calls (*n*) to select the proper elements for my header and position each where I want the header elements to appear. Then I save the entire works back out to the section records on the disk.

When I print, the master printing ruler document calls the section records one at a time. When I start the Formatter running, it asks at the third parameter, USE MAILING LIST? N and I enter the YES at this time. The sections, as they are printed, will call the headers as they are needed, from the mailing list document. By using this approach, it is very easy to make use of section headers along with running headers. The .HE command in the master printing ruler document is not used at all. However, the .FO command serves to insert page numbers throughout the entire book.

(Continued on Page 12)

"The Far Right Stuff"
by Mike Perry, et al.
Berkeley, California

Reprinted from the March/April 1985
Forth DIMENSIONS.

There are those among us who are attempting to advance by only one generation of computer technology. It is just such an egregious lack of imagination which has put us into the dire straits in which we find ourselves today. Why stop with the fifth generation, when logic has already taken us as far as it can? It is time to throw off the Markov chains which have bound our thinking for so long. Let us retain our integrity and not sink into a morass of floating points.

In short, I propose that work begin on a sixth generation computer, when the knowledge-based expert systems of the fifth generation have exhausted the possibilities of mere rational thought, all eyes will turn to the potential of the irrational. It is not too early to develop intuitive machines, which I like to think of as ignorance-based assumption systems. They will, of course, use irrational logic gates, which are based (naturally) on the natural number base (2.7 et cetera). Ordinary logical reasoning can be compared to a chain: an inference is only as strong as its weakest link. Intuitive reasoning is more like a rope: no one strand is very strong, but when tightly coupled into a rope, conclusions are bound to be inexcusable. The possibilities are imponderable.

There are many advantages to sixth generation machines. For one, ignorance systems will consume very little power: as everyone knows, Knowledge is Power. They can be built from some well-known illogic gates, such as the ignorance gate (don't confuse me with the facts), the obvious gate (as any fool can clearly see) and the sure-thing gate (it's in the mail). As all of these are output-only devices, no input will be required, greatly simplifying system design. This will lead of replacement of the old computer adage "Garbage In, Garbage Out" by simply "Garbage Out."

Performance measurement will also be affected. Instead of MIPS (millions of instructions per second) or KLIPS (kilo logical inferences per second), the sixth

generation machines will be measured in DUPES (deca unreasonable predictions each second).

New languages will be required. As demonstrated below, the older languages were very limited.

ASCRAMBLER The first languages were an improvement over their predecessors, and helped to enhanced problems to truly macroscopic proportions.

FORTUITOUS Filled an array of real needs at first, but it was beginner's luck. Neither its backus nor anyone else would recognize it now.

GOBBLE Government Organized Big Business Language was a real turkey, to wordy and difficult to pronounce. Its users were renowned for the weight-lifting abilities.

GARGOYLE BEGIN GARGOYLE := ACADEMIC. It had a Horrible appearance, AND was cursed time and again. It was super-setted by RASCAL in the END.

SIMPLE 10 Basically too complex and hard to use. 20 Abused the powerful COME-FROM construct.

LISTLESS (and (used Polish notations) (put old parentheses on a garbage heap)).

RASCAL While not without its Wirth, it was unforgiving and not really modular, too.

FROTH At first, even its starry-eyed followers thought it would not float, but when it came un-Moored they could see that it did. On the stack it pushed more than once has been.

GOSSIP A classy language which supported chatting in both back-fence and window modes. Some objected to passing this sort of messages, feeling that some facts were better kept private.

SEA (often confused with OCEAN; it is actually rather unquiescent;) SEA was one inspiration for the saying, "I'm elegant, you're terse, he's unreadable."

MONOLITH To make RASCAL suitable for numerous tasks, new types of checking were implemented for a program's definitions. A recent export of great import.

ADORD The official language of the Department of Redundancy Department, who decided to Boole their resources. Before it is valuable, it will be dated.

PROLONGED If the inference of some experts are proved correct, then its deductions may eventually lead the way toward superficial intelligence.

OXYMORON Allows an Ocean of disputer processors to run the RISC of producing abysmal results. Sort of a contradiction in terms.

To solve the problems caused by sixth generation machines, new languages will have to be created.. Here are a few possibilities:

GUESSWORK Can be used to implement functions like What-Did-I-Mean?

ABSURD Uses backward inference to reduce the solution until only the problem remains. Given a hypertheoretical situation, it expands the problem until it interferes with the solution, thus disproving the result.

As grand as these schemes may sound, let us not think that the sixth generation is in any way the ultimate. Already the characteristics of the seventh generation are becoming apparent. Taking the chance of combining the best features of irrational logic with the technology of virtual machines, we will at last see real imaginary computers. This will lead to the replacement of the old saying, "Never trust a computer you can't lift" by the new adage, "Never trust a computer big enough to see."



On The Lighter Side

The other day I received a letter that I would like to share with the rest of you:

MEMO TO: MSP 99 Users' Group
FROM: Siouxland 99'ers
Tom Graff Managing Editor
SUBJECT: Newsletter Exchange

"Just the other day our Secretary/Librarian Bob mentioned to the other officers of our club that he would like to a) spend a night with Barbi Benton; b) win the Iowa State Lottery; and c) get a subscription to your newsletter.

"Well, we all like Bob, and he works real hard at being a good librarian and secretary to the Group, so we all decided to see what we could do to make his dreams come true. Rory volunteered to "get in touch" with Barbi; Ike offered to contact Iowa about rigging the Lottery; and I volunteered to write this note asking you to help us with our quest.

"Naturally, it would be nice if we just came out and "bought" a subscription to your newsletter, but Rory has run up some outstanding phone bills (he keeps getting disconnected), and Ike says what little funds we have in our treasury will be needed to defend ourselves in a "racketeering" lawsuit. I am hoping you will accept a simple "exchange" of newsletters and the reward of knowing you're helping to make Bob's dream a reality as payment enough.

"P.S. I don't suppose you could have Barbi deliver it?"

Well, it's not everyday I receive a letter that is such a joy to read. I think that we can do our part in making Bob's dreams come true. You can consider the Siouxland 99'ers to be now on our newsletter exchange mailing list. However I'm not sure Barbi will be able to deliver it. She's usually much too busy just licking my stamps. I'll see if Loni Anderson can drop it off on her way home though.

XBASIC REQUIRED

DON'T LOSE THAT LINE

How many times has this happened to you while editing a line? Instead of pressing FCTN 2 (Insert) you accidentally press FCTN 3 (Erase). What do you do? The main thing is DO NOT PRESS ENTER! Instead, press FCTN P (") once and then press ENTER. The computer will read it as an incorrect statement (unmatched quotes) and default back to the original statement.

USING PRE-SCAN

One of the major drawbacks to the TI 99/4A computer is its speed, or lack of it. Once you've loaded your program and typed RUN there is a period of waiting while the computer checks your program and sets aside memory space for various things like variables. This is called pre-scanning. However, if all these items are included in the first few lines of the listing, the computer must still check the entire listing before it can run the program. Unless you tell it not to.

You can do this with the use of two simple commands: !@P- and !@P+. The first will turn the pre-scan off while the second turns it back on. Note that the pre-scan is on when the computer is powered up. The following must be included in the section of the program that is pre-scanned: all variable and array names; all DEF statements; all CALL statements; the OPTION BASE statement; and at least the first DATA item. The majority of this can be placed at the beginning of a listing, then turn the pre-scan off. You can turn it back on again if you need to and you must before any subprograms are run.

Here's a little program that will printout return address labels. It requires an RS232 PIO port.

```
100 REM filename: ADRL/PIO/R
110 REM purpose: Print 3 Up
    Return Address Labels
120 REM author: rdac 1/85
130 REM -----
140 CALL CLEAR
150 FOR SCR=1 TO 12 :: CALL
```

```
COLOR(SCR,16,1):: NEXT SCR :
: CALL SCREEN(7)
160 !*** Setup name to be pr
    inted *****
170 NA$=CHR$(27)&CHR$(87)&CH
    R$(1)&"NAMEum"&CHR$(27)&CHR$
    (87)&CHR$(0)
180 AF$="A Name Here"
190 AD$="9999 Anystreet"
200 CS$="Anytown, MN 55424"
210 DISPLAY AT(2,1)ERASE ALL
: "RETURN ADDRESS LABEL PROGR
    AM" :: DISPLAY AT(3,1): "Uses
    3-up 1x2-5/8in. Labels":RPT
    $("-",28)
220 DISPLAY AT(10,1): "printi
    ng labels for:" : NA$:AF$:AD
    $:CS$
230 DISPLAY AT(22,1):RPT$("-
    ",28): "PREPARE PRINTER FOR P
    RINTING": " press any key to
    continue "
240 CALL KEY(5,KY,ST):: IF S
    T=0 THEN 240
250 CO=0
260 OPEN #2:"PIO"
270 PRINT #2:CHR$(27);CHR$(5
    6):: PRINT #2
280 FOR I=1 TO 9
290 CO=CO+1
300 PRINT #2:TAB(3);NA$;TAB(
    31);NA$;TAB(58);NA$
310 PRINT #2:TAB(3);AF$;TAB(
    31);AF$;TAB(58);AF$
320 PRINT #2:TAB(3);AD$;TAB(
    31);AD$;TAB(58);AD$
330 PRINT #2:TAB(3);CS$;TAB(
    31);CS$;TAB(58);CS$
340 IF CO=9 THEN 350 ELSE 36
    0
350 PRINT #2:CHR$(27)&CHR$(1
    0):: GOTO 370
360 PRINT #2 :: PRINT #2
370 NEXT I
380 CLOSE #2 :: CALL CLEAR
390 END
```

When entering a listing with many similar lines, you can save time by using REDO (FCTN 8). Just type the first line and hit ENTER. Then, press REDO. The line will reappear with the cursor on the first digit of the line number. Change the line number and modify the rest of the line as usual.

Ever wanted to disable the QUIT key (FCTN +/-)? Here's how. Just type:
CALL INIT :: CALL LOAD(-31806,16)
The Quit key will not work until you turn the computer off and back on again, or until you use:
CALL INIT :: CALL LOAD(-31806,0)
Note that you can use this within a program listing as well.

MAX-RLE PICTURES

(This is a summary of the instructions on how to use MAX RLE. This information has been available previously, but it is summarized here by popular request.)

RLE stands for Run Length Encoded. It is a program for preparing and viewing digitized pictures, both artwork and photographs, sent between computers over phone lines using a terminal emulator such as Fast Term. Many computers use this technique with the VIDTEX terminal emulator protocol which permits viewing pictures on-line. For the TI 99/4A at present, pictures can be viewed off-line only, but pictures can be exchanged with other brands of computers. The program supports four different formats - both TI-ARTIST and GRAPHX formats, as well as Display-Fixed 128 (the usual format used in other computers) and Display-Variable 80 format.

LOADING MAX-RLE - The program is loaded using the E/A module or its equivalent, Option 3 - Load and Run. The filename is usually MAX-RLE (this may have been changed on your particular version) and the program name is START. The MAX-RLE title screen will then appear asking for the name of the picture file you want to load.

RUNNING MAX-RLE - At the title screen, you have two options - you can load a picture or you can catalog a disk.

LOAD A PICTURE - Just type in the filename, for example, DSK1.PICTURE, and press ENTER. Whatever format the picture is in, the program will recognize it and load it. (NOTE: For TI-ARTIST files, omit the "P" and "C" at the end of the filename - the program provides these automatically.) You will then see a grey screen for a short while as the picture loads. The picture will then appear all at once on the screen.

CATALOG A DISK - Just type DSKn.", where n is the drive number. Be sure you include the period.

Once the picture appears there are three options open to you - You can return to the title screen, print the picture, or save the picture to disk.

To return to the MAX-RLE title screen just press ENTER. This removes the picture from memory.

To printout the picture, press P. The default setting of PIO.CR will appear. If you are using something other than a parallel printer port you can now change this designation. Your printer must be compatible with the GEMINI-EPSON family in its handling of dot graphics.

To save to disk, press S. The default setting of GRAPHX will appear on the screen. To save in a different format, press the space bar until the format you want appears. Then type the filename you wish to save to.

SENDING PICTURE FILES - Generally, pictures to be transmitted should be saved in DF/128 format and uploaded with XMODEM transfers. This is the format used by other systems. Pictures can be sent in DF/80 format using ASCII (text) transfers, but they lack error checking in transmission and a noisy or weak connection can ruin the resulting picture.

PICTURES ON COMPUSERVE - Pictures readable by MAX-RLE can be found on CompuServe in the TI Forum Data Libraries, the PICSIG, the ARTFORUM, and the CB simulator area. They are also appearing on many BBSs.

MAX-RLE is available from the MSP 99 Library if you do not already have it.

Reprinted from the PUNN Users Group newsletter March 1987.
(Author unknown)

 WHY ARE FIRETRUCKS RED?

Well, firetrucks have four wheels and eight men. Four and eight make twelve. There are twelve inches in a foot, and a foot is a ruler. Queen Elizabeth is a ruler, and it's also one of the largest ships on the sea. The sea has fish and fish have fins. The Finns fought the Russians once and the Russians are red. That's why firetrucks are red, 'cause they're always rushin'.

If you think this is illogical, you should hear some of the excuses people give for not participating in club activities.

Reprinted from the LA 99ers TopIcs

Tips 'n Thoughts...cont from Page 7

Alternately, you may wish to number pages by the section. This can be done quite easily by removing the .FO command from the master print ruler and putting it into each section instead. In this manner, the Formatter can print pages in the format 1-1, 1-2, and so on with the first number being the section number. This is often done in texts and in technical manuals. If a section or chapter runs two or more records, then use the command in each record, but follow it right away with the .PA command. This allows you to set the page numbers that will be printed in the section footers.

I have used the mailing list method almost exclusively for headers and footers because it is so versatile. It sure is fun to watch the system print out this completely formatted manual when I am all through, and the books we have published in this manner have gotten some nice comments.

I guess when you write manuals for a living, it gets to be a challenge to see how professional looking you can get with the home system. This has sort of been the push behind learning the capabilities of the TI-Writer system to the degree I have.

I promise the next column will be on a different subject, though!!

WANT ADS_____

FOR SALE
Texas Instruments Thermal Paper
(8 1/2" x 100") P/N 972603
Call Mel.....378-2149

WANTED
Used TI equipment and software for
youth computer group.
Call James.....429-6081 (Home)
429-1518 (Work)

```

1 !BEERSTEIN WAVE BY STEVE MICKELSON (9/86)
100 CALL CLEAR :: CALL CHAR(42,"3F7CDDDDDDDD7C3F"):: J=-1
110 FOR B=1 TO 28 :: CALL SPRITE(#B,42,5,B*4,10,0,B*1):: NEXT B
120 FOR B=1 TO 28 :: CALL MOTION(#B,0,B*J):: NEXT B
130 J=J*-1 :: GOTO 120

```

**MSP 99 Users Group
P.O.Box 12351
St. Paul, MN 55112**

TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY