

**SASKATOON  
COMPUTER  
CLUB**

**TEXAS  
INSTRUMENTS**

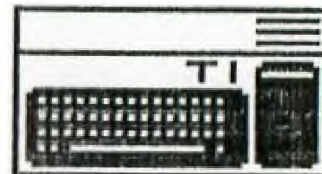
*APRIL - MAY, 1988*

**FROM:  
P.O.Box 7925  
SASKATOON  
SASKATCHEWAN  
CANADA  
S7K 4R6**

**FIRST CLASS MAIL**

**TO:**

SASKATOON  
TEXAS INSTRUMENTS  
COMPUTER CLUB



We discuss and review new products for the TI99/4A while providing technical support for any problems that a member may have. We also support a software library and have regular contacts with other groups in Canada and the United States. Our membership fees are very reasonable: \$10.00/single or \$12.00/family. If you would like to become a member, or require more information, contact any member of the executive.

EXECUTIVE COUNCIL:

PRESIDENT: - - - - - Harry Caruk - - - - 384-6321  
VICE PRESIDENT AND SECRETARY: Steve Zabarylo - - - 931-7812  
TREASURER AND LIBRARIAN: - - Mike Trimble - - - - 242-6215  
EDITOR: - - - - - John Thomson - - - - 244-1394

MAILING ADDRESS:

P.O. BOX 7925, Saskatoon, Sask., Canada S7K 4R6

STICC DISCLAIMER:

We are a small, nonprofit club here in Saskatoon. We never have, or ever will, consider our newsletter a professional medium in its field. Therefore, with this consideration, the views presented in the articles, by various authors, may not be the views of the newsletter committee, the STICC executive, or its members. On this basis, this club cannot be held responsible for errors, omissions, views, or copy infringements, as presented in the articles.

MEETINGS:

General meetings were normally held at 7 PM on the first monday of each month at Kelsey Institute, Idylwyld Drive at 33rd Street, in room 132. There is free plug-in parking behind Kelsey in the student parking stalls; if entering from this parking lot, use door #10, otherwise the front door closest to Idylwyld and 33rd can be used.

It was decided at the April meeting that commencing in September, 1988, we will change our meeting night to the second monday of the month, or the third monday if the second monday is a holiday. This has yet to be finalized, depending on availability of a suitable room at Kelsey. If we don't call you, please telephone one of the executive to confirm the meeting night and room.

\*\*\*\*\*

The following article is contributed by Steve Zabarylo, a STICC member.

### MEMORY EXPANSION

How many more ways will we find to add memory to the venerable 99/4A? We've put it in the console, PEB, and all manner of stand-alone devices including the speech synthesizer. Until recently all added memory was placed on the eight bit data bus of the system. Only system ROM and scratch-pad RAM could be addressed a word at a time (two bytes, or 16 bits), without data passing through the 16/8 data bus multiplexer. One of the penalties of using the bus multiplexer is the time required for the circuitry to do it's job. While it is placing data words from the microprocessor on the eight bit bus one byte at a time, or assembling words from consecutive bytes to be presented to the microprocessor as words, the microprocessor must spend some time doing nothing. The fancy term for this non-activity is referred to as "wait states". If memory expansion could be placed on the data bus similar to the system ROM, so that it was addressed on word instead of byte boundaries then wait states could be eliminated for the operations involving that memory. As the console already has circuitry that decides when to use or not use wait states, then altering that process is achieved with the addition of some simple logic. With the recent availability of 16k by 8 bit and 32k by 8 bit static RAM chips at affordable prices the home experimenter has a whole new series of "memory expansions" to attempt. What follows is a review of my attempts at two versions of "memory expansion on the sixteen bit bus".

My first 16 bit bus project followed the design offered by Mike Ballman. This version uses two HM62256LP-15 32K by 8 bit static RAM chips. Construction is only slightly more complicated than earlier memory projects, due to working directly on the crowded mother board. Removing the ROMs and installing sockets made the job much easier but this requires access to a good desoldering station, or very patient work with the solder wick. Because of close clearances with the RFI shielding, extra care must be used to ensure no extended chip leads are shorted out when things are reassembled. While things are apart, do consider installing some wires to bring out to an external reset switch. I also cut trace 1 on the grom port extender so that my reset switch works even when a module is installed.

You can't imagine my pleasure when I first booted my newly revised console and typed SIZE from the \*READY\* prompt, and extended basic told me all that memory was there. A quick check with Advanced Diagnostics confirmed that the memory was fully functional. The question of software compatability is the proof of the pudding though for this project. The software I use most, TI Writer, has run perfectly so far. There is a distinct improvement in the wordwrapping feature. No more dropped letters, a severe annoyance before. All memory operations, such as searches and moves, seem to take half the time they did before. Multiplan seems to be quite happy with the new memory. The extra speed has made the spreadsheet much less of a chore to use. For the average typist, the skipped letters on data entry happen far less often and moving around on the sheet is far faster. And so the story goes with other software that I try, such as DM1000, PRbase, TI-Artist, Advanced Diagnostics, and Disk Utilities. Then we struck out with Fast-Copy. This program uses it's own routines in place of the code that resides in the disk controller. As disk drives are very timing dependant devices the timing loops this code takes for granted in normal memory just don't work from this new memory. Other programs that use their own code to operate the drives seem to exhibit the same inability to access the drives. Games like TI-Runner that load a series of screens from disk are unable to function. The timing problem is manifest in other games in a different way. Computer Wars, which is already far too fast for me, becomes insanely fast. Many of my other favorite games such as Jumpy and Parsec seem to work just fine.

\*\*\*\*\*

## "MEMORY EXPANSION" continued:

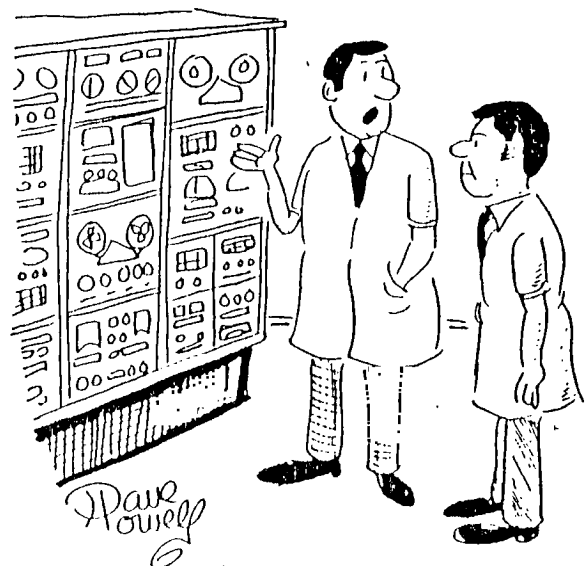
So it seems that in order to become a fully compatible hardware 'option' a way must be found to bridge those software problems, perhaps in the form of a fast/slow switch for the mod, or less likely, versions of the software modified for proper timing. My one attempt at a fast/slow switch did not work. What I may resort to, is adding one of my earlier 8 bit memory expansions and change speeds by powering up one expansion or the other. This may turn out to be the easiest and cheapest option.

Lastly I would like to cover an alternate 16 bit bus expansion method shared with me by Dave Hamm from Nova Scotia. His method uses four of the 6264LP-15 chips plus one more chip for the address decoding than the previous method using the 32k byte chips. While there is a saving in the cost of memory (14.50 versus 40.00) almost twice as much work is involved in the installation. Another incentive for springing for the larger chips is the likely eventual development of projects to take advantage of the presently unused space (32k bytes of the 64k added with the two chips is not used).

So there you have one rosin sniffers experience with yet another hardware hack to the old orphan. By all means, don't be afraid to try it, and if your software diet is similar to mine, you will find the results well worth the effort.



"No matter what the project he's working on is supposed to be—it ends up being a light dimmer."



"I feed my problems into this computer and all it does is multiply them!"

\*\*\*\*\*

The following tips come to us via the Houston Users' Group:

TI-WRITER TIPS #1  
 - by Bob Seddon -  
 EDITOR MARGINS VERSUS  
 FORMATTER MARGINS

There are two kinds of margins in TI-Writer: Editor Tabs & Formatter Dot commands.

EDITOR TABS

When you create text with the Editor you use margins called Tabs. Tabs are set via CTRL c (PROMPTS), t (Tabs), Enter. This sequence of keystrokes makes the Tab Line appear across the top of the screen. You can type over the Default settings at 0 and 79 and reposition L and R to make on-screen tabs any width within that range.

PRINTING WITH THE EDITOR

You can print text created in the Editor with the Editor itself by CTRL c (PROMPTS), f (FILES), pf (Print File), Enter. This sequence of keystrokes prints text with margins equal to the tab settings; the printer output resembles the screen.

However, there are advantages in NOT using this method to print. If you use the Formatter instead of the Editor you can print lines longer than 80 spaces. You can automatically number successive pages. You can put Headers at the top of each page, Footers at each bottom. You can make the R margin flush with the .FI;AD command. You might want to use the Ampersand to underline, the Each to Print Bold. You can double space and set page length.

PRINTING WITH THE FORMATTER

To print through the Formatter you must Save the file, Exit the Editor, Load the Formatter, reLoad the file, and then print. The Formatter prints the file according to the Dot Command instructions.

FORMATTER DOT COMMANDS

Formatter margins are also set in the Editor, but not the same way as the Tabs. Instead, Formatter margins are typed in (usually on line 0001) as Dot

Commands (.LM n;RM n). Dot Command margins (if present) override Tab margins when text is printed through the Formatter. If there are no Dot Commands the file will print out according to the Tab margins. The Formatter follows the Dot Command instructions but does not print the Dot Commands as it does text. The Editor, on the other hand, not only ignores Dot Commands but also prints them just as it will any other text, since it cannot make the distinction between Dot Commands and regular text.

Quite often you will want to print your text with margins EXACTLY the same as on-screen. There are at least two reasons you might want to do this:

(1) HYPHENS

If you pack in as much text per line as possible you will want to break words and hyphenate them; if you do so, the final printed output must break the words at the same place you did. Otherwise, your text will take on the appearance of this particular sentence.

(2) MULTIPLE COLUMNS

If you create text with narrow columns so that you can put several parallel columns on one page you need to count the EXACT number of lines. Line numbers down the left column give you this number (minus the lines devoted to printer commands) if your on-screen equals your printed work. (NOTE: see box at end of article about the advantages of narrow columns.)

There is a trick you must use to make your on-screen work created in the Editor resemble the printed output of the Formatter so that each resembles the other line-by-line.

R TAB ONE # HIGHER THAN .RM

If you use the Formatter you must set the R Tab one digit higher than the setting of the .RM dot command. The difference between the Formatter and Editor is that the Formatter will print ON the .RM column. The Editor prints UP TD (but not on) the R Tab.

TAB SETTINGS: 0 & 31  
 SPACES USED ON-SCREEN: 0 - 30  
 (31 ACTUAL SPACES OCCUPIED)  
 DDT SETTINGS: 0 & 30  
 COLUMNS PRINTED ON: 0 - 30  
 (31 ACTUAL SPACES OCCUPIED)

In this article I set the Editor margins at 0 and 31 to fit three columns on the page. The Formatter settings are at .LM 0;RM 30. You can see that the columns printed at a width of 31, not 30.

123456789 123456789 123456789  
 L....T....T....T....T....T....T

When counting, remember to begin ON zero: call the .D a 1, 1 a 2, 2 a 3, etc. R follows the last T. We cannot print it here for the very reason being discussed! Counting the number of spaces used in each line on printed work is not difficult. Nor is it hard to count spaces used on screen. It is even easy to memorize the rule for making Editor margin width one character longer than dot command margin width. The one frustrating thing about this whole business is allowance for a L margin on zero!

Instead of making .RM one digit smaller than the R tab setting you have the option of achieving the same effect by making the .LM one digit greater than the L Tab of the Editor. The option of using zero as a L Tab adds confusion to this issue because you can also set .LM at zero; thus, this tactic is of no particular benefit for you.

SPACE(S) AFTER . : ? !  
 REFORMAT VERSUS .FI

When you Reformat, the Editor packs in text according to its own set of rules, rules different from the Formatter .FI command. If you leave only one space after a period, the Editor's Reformat command will NOT increase the spacing to two places. The Formatter, on the other hand, ALWAYS leaves two spaces after periods, whether you want it to (at sentence ends) or not (after initials).

\*\*\*\*\*

We can prevent the Formatter from increasing the single space after the final dots of initials, abbreviations, etc. by putting a carat between such dots and the next letter.

Similarly, we need to force in two places after :, ?, and ! If you merely leave two spaces after each one the Formatter will reduce your two spaces after each of these down to one unless you follow them with a carat, then the space. Optionally, you can key in two carats (and no space).

---

#### SPACES LEFT AFTER:

	period	initial	?!:
Editor	1	1	1
Reformat			
Formatter	2	2	1
.FI			
Remedy	space twice	carat once	carat twice

---

#### FORCED IN CARRIAGE RETURN

The Formatter makes a decision to Wrap based on the R tab setting and whether a word (or ANY group of characters) occupies or exceeds that setting. Usually the last PRINTED character in a paragraph is a period and if it falls on the last occupiable position (R tab setting minus one) you must be careful where you place the carriage return.

- (1) If you space once after the period, then Key CTRL m, there is no problem.
- (2) If you cursor down below your text, then Key CTRL m, there is no problem.
- (3) If you key CTRL m in the position directly following the period, the last word in a paragraph will not fit at the end of the line and will drop to the next line.

When it drops, you notice that it SHOULD fit, even when you account for the space before the word and the period following. The Editor Wraps the word around to the next line because it treats the carriage return

following the period as part of the word, even though the carriage return is not a printed symbol.

If this happens to you, you must break the text after the period and before the carriage return (CTRL g), then Reformat. The word will now NOT wrap to the bottom line. The carriage return also moves up to the original line.

#### FORCED IN FORMATTER COMMAND

A similar problem occurs when you precede a word with an ampersand, carat, or @. Let us consider the ampersand which is, of course, a Formatter command to underline any word it precedes. The Formatter .FI Command ignores the ampersand and packs in Text as though the ampersand were not there. Unfortunately, the Editor treats the ampersand as a regular character when Reformatting and, so, will make a decision to Wrap a line based on the presence of it within a line of text, just like the carriage return. This anomaly makes it difficult to create a line of text which appears on screen exactly as it will print.

There is a technique to insert these codes in front of any (or even every) word on the line. Unlike the carriage return which FOLLOWS a word, a Formatter command PRECEDES it, making the previous technique impossible. Turn off wordwrap with CTRL O. This turns your cursor into a hollow rectangle. Move the cursor to the letter before which you want an &. Key in Insert (FCTN 2). Key in the ampersand. Everything right of it will move right one column. This is the only way you can make a character appear ON column R.

You can only insert one such ampersand per line using this trick UNLESS YOU RESET THE EDITOR R MARGIN TO A HIGHER NUMBER. If you insert an additional character anywhere else on the line and do not first increase the R margin the last character on the line will vanish. You can precede EVERY

word in the line with a non-printable character so long as you increase the R Tab enough so that all text and all codes fit on that line. The only restrictions which apply are that you may not mix text and code such that you exceed 80 spaces; nor can you Reformat afterwards.

#### THE NARROW COLUMN ADVANTAGE: NO WINDOWING

I find it convenient to set on-screen margins so I can see all text without Windowing left and right. Since I also like to leave the four-digit column numbers on the left side of the screen at all times, the highest possible R margin setting is 34. (34 is off screen, but Wordwrap causes text to occupy 33 by briefly Windowing right, then left, when you type on 33 itself.)

You can turn off the column numbers (FCTN O) to see six more columns of text (4 digits and 2 spaces) which lets you set the margins at 0 and 39. You can even set R at 40 (which is off screen) so that Wordwrap will cause text to occupy 39; this makes a REAL 40 column screen. Naturally, the R Tab can be reset anytime to Reformat to any desired margin width up to the on-screen maximum of 0-79.

#### 79 COLUMN SCREEN

I am sorry to break the news to you that you only have a 79 column screen, not the 80 column screen you thought you did! When in Wordwrap the Editor does not let text occupy the column of the R margin (on screen or when printed with the Editor). If you create text with Wordwrap on, the Editor alone cannot print 80 columns: its extremes are 0 and 79, and 79 is not printed on. The arithmetic is tricky because of the presence of the zero. If your Editor margins are on 0 and 79 you can only print 80 column text IF your dot commands are set at 0 and 79! AND IF you use .FI; nor will on-screen equal output.

TI RUNNER Hints /Reminders:

You can skip levels in TI Runner by pressing FUNCTION 5 at the beginning of the level you wish to skip. There are situations on some levels where you (your white sprite!) can fall into a hole from which there is no escape and in which you can not even commit suicide by using the joystick. In that event, press 6, the suicide key. When any screen is loaded you can go directly to any other screen. Notice your white man flashing. Press SHIFT 3. Your white man will stop flashing. Enter any two digit number from 01 to 50 to go directly to that screen. Need a break part way through a screen? Press P for Pause and C to continue. I thought I was doing great to reach level 21, until I discovered there are 50 levels in this game! By the way, we have a program that allows you to customize the game by editing screens, as well as a fairware version with different screens.

DM1000 Disk Catalog Hint/Reminder:

If you want a printout of a diskette catalog on a piece of paper small enough to attach to the front of a diskette sleeve, and you have a lot of files on the diskette, you may wish to print the catalog with tiny print. There are several programs available to print disk catalogs in various forms, and we may review them in future newsletters, but if you are in a hurry and you don't want to locate, load, and figure out how to operate another program just to do one or two disk catalogs, then you can do it directly from DM1000, provided your printer supports compressed subscript printing.

When DM1000 has loaded, and you are prompted for options 1,2,or3, press FUNCTION 3. You will be prompted to "Enter List Device". The default of "PIO" will likely be there. Press ENTER. You will then be prompted to "Send Control Codes (Y/N)". Enter Y. Enter the following string of numbers, including the spaces and the asterisk.

15 27 49 27 83 01 \*

These codes will select compressed character mode, set the line spacing to 7/72 inch, and select subscript mode. Of course, you can vary these as you like or as you need to for your particular type of printer. You will then be prompted "Save to Disk (Y/N)". If you answer Y, and the write protect sticker is removed from the disk, these settings will be saved as your defaults. I would suggest using an extra backup copy if you choose to do this. Otherwise answer N.

To test this out, choose option 2 (Disk Utilities), option 1(catalog), disk 1. With the printer on-line, press FUNCTION 7. The printout should look something like this:

File Name	Free	Used	358
-----	Size	Type/No.	20 P
AB	33	PROGRAM	U
AT	22	PROGRAM	U
CHARA1	8	PROGRAM	U
CHARA2	8	PROGRAM	U
BB	2	PROGRAM	U
BP	17	PROGRAM	U
EA	8	PROGRAM	U
ED	33	PROGRAM	U
EE	19	PROGRAM	U
FD	33	PROGRAM	U
FP	18	PROGRAM	U
LL	10	PROGRAM	U
LOAD	32	PROGRAM	U
MD	2	PROGRAM	U
ME	33	PROGRAM	U
MH	30	PROGRAM	U
MB	10	PROGRAM	U
BL	10	PROGRAM	U
UL	4	PROGRAM	U
UTIL	33	PROGRAM	U

SOFTWARE NEWS:

INSCEBOT Inc has recently come out with a new data base program called TI-BASE. Our club should be receiving a demo copy in the near future, and then it will be demonstrated at one of our monthly meetings, possibly in September.

To quote from their promotional brochure: " TI-BASE is the place to collect and organize your data. Record definitions are customized to your exact needs. Data may be freely interchanged between all types of records. Simple structured command language allows an easy implementation of needed functions. Address lists, check-books, business applications, all easily performed and maintained on diskette by TI-BASE.

- DATA DEFINITION: character, numerical, date, literals
- SORTED RECORDS: sequential, "find"
- MATH CAPABILITY: arithmetic, trig, logical, boolean
- DISK DIRECTIVES: initialization, catalogs, file copy, file deletion
- COMMAND LANGUAGE: nested command files, structured directives
- LOCAL VARIABLES
- ON-LINE HELP
- CHARACTER MANIPULATION "

Group orders will be discounted 20%: from \$24.95 to \$19.95 . This just might be the data base program some of us have been waiting for.

**SPECIAL  
STICC  
MEETING  
TUESDAY, JUNE 7  
AT THE HOME OF:  
STEVE ZABARLYO  
106 0" BRIEN CRES.**