BUG BYTES

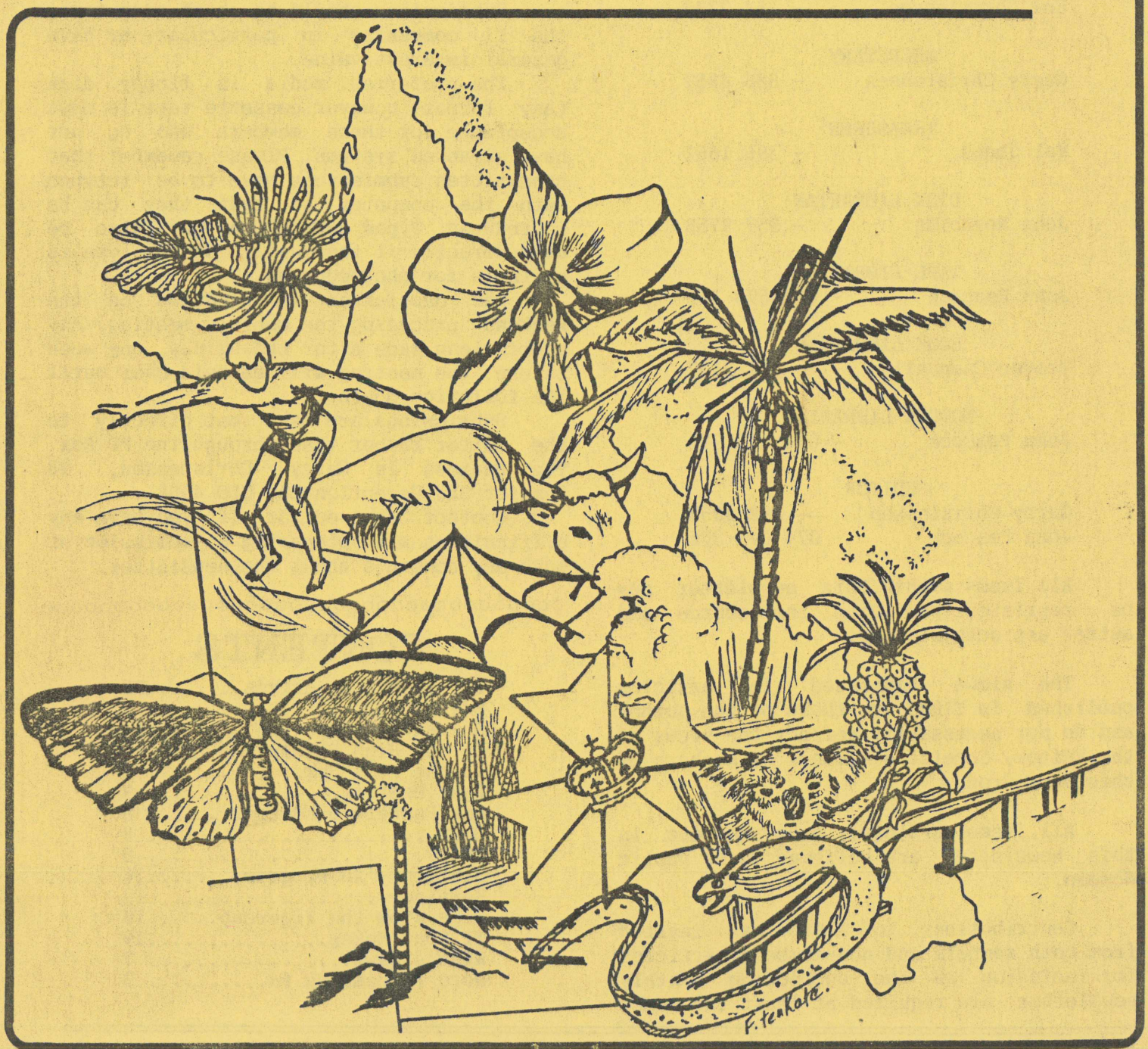
NEWSLETTER

of

TIBUG

TI - 99/4A - BRISBANE USER GROUP INC
P.O. BOX 3051
CLONTARF MDC, QLD AUST 4019

JULY 1992



COMING MEETINGS
31 JULY & 28 AUGUST

7.30 p.m.
EAST BRISBANE STATE SCHOOL
CNR WELLINGTON RD. AND
STANLEY STREETS,
EAST BRISBANE.

COMMITTEE

PRESIDENT
Ralph Nielsen - 378 1237

VICE-PRESIDENT
Col Christensen - 284 7783

SECRETARY
Garry Christensen - 888 4857

TREASURER
Val Jones - 396 1662

DISK LIBRARIAN
John Reynolds - 357 9758

TAPE LIBRARIAN
John Peacock - 074 673 376

BOOK LIBRARIAN
Trevor Campbell - 351 3107

MODULE LIBRARIAN
John Peacock - 074 673 376

EDITOR
Garry Christensen - 888 4857
John Peacock - 074 673 376

The views expressed in articles published in TIBUG are those of the author and do not necessarily reflect the views of the Editor, Committee Members or Members of this User Group.

All items, articles, programs etc in this Newsletter are believed to be public domain.

Contributions to TIBUG are invited from both members and non-members. Articles for inclusion in the succeeding monthly newsletter are required at least two weeks before the monthly meeting and may be included in that newsletter at the discretion of the Editor. If you have a disk system, please supply script on disk with diagrams separately on paper and as clear and black as possible to facilitate photocopying.

Most original articles by members of TIBUG in this newsletter are on disk and are available to other User Groups on request.

Submissions of articles, reviews, comments and letters from mebers is encouraged, however the editor would ask that members keep the following in mind.

Submissions should be about computers, the TI community in particular, or have general interest value.

The preferred media is floppy disk (any format) however cassette tape is most acceptable for those members who do not have expanded systems. Please remember that handwritten submissions have to be retyped into the computer so that they can be reproduced. Typed submissions can also be used directly if the quality of the type is suitable for photocopying.

The newsletter is produced on the weekend preceding the monthly meeting. Any submissions made after the Friday, one week before the meeting will be held over until the following month.

Submissions are best sent directly to the Editor rather than through the PO Box. The address is Garry Christensen, 18 Zammitt St, Deception Bay QLD 4508.

Contact the editor if you have any difficulties with preparing a submission or have any comments about the newsletter.

oooooooooooooooooooooooooooooooooo

CONTENTS

EDITORIAL

Guess who? Yep, I couldn't let go that easy so I'm here for another month. Actually, John Peacock was all set to take over but the death of his mother-in-law threw the family into disarray. I'm sure everyone joins me in extending our deepest sympathies to you, John.

Well, this is going to be a quicky one this month. I have just started classes at the QUT. You would not believe it. Monday and Wednesday evening, Friday afternoon and evening and Saturday morning. I feel like I live there. Add to that a couple of nights a week study and the requirement for a minimum of 720 hours of clinical experience in the next 11 months (I've still got to work to) and I'm sure that you will see why I will not be doing too much on the computer for a while.

Now you may think that I am crazy for even taking that much on, and you are probably right but Tracey and I have also decided to start a business as well. I haven't worked out when I am going to find time for that yet but I guess that it will all work out.

As I said, short and sweet this month. Next month I promise you a new editor.

oooooooooooooooooooooooooooooooooo

WORD PROCESSING #4

by Col Christensen

Last month's article covered a number of dot commands placed in a text file and used as instructions by the Text Formatter. Some of these can have numerical parameters following them. Reviewing one point, a dot command such as .LM8 has the absolute value, 8, and sets the left margin to the 8th print column whereas .LM+4 with a relative value of 4 sets the left margin at 4 columns greater than the base setting. All relative values in dot commands refer back to the base setting of that command except for .IN+n which refers to the left margin base setting.

HIGHLIGHTING AND SPECIAL EFFECTS.

Certain characters are used by the Text Formatter as flags to effect some print output features. The characters in question are the & for underlining, @ for overstriking and ^ for required space. (ampersand, at and circumflex).

REQUIRED SPACE with ^

The circumflex (^) tells the Text Formatter that words joined by it are to be treated as one word during underlining, overstriking, filling text or adjusting text. For example, you may want a group of words such as 24 January, 1992 to appear all on one line and not split partly on one line and part on the next. Just use the circumflex to replace each space underlined. But if you wish to underline a group of words there are two ways to go about it. The first is to use the & symbol before each word in the phrase, e.g. &The &Storm &King. Then each word will be between the words. The Text Formatter will treat the whole as a unit and print it on one line with a space character replacing each circumflex. The two paragraphs following show how the required space is used during underlining and overstriking.

UNDERLINING with &

Wherever the Text Formatter encounters the & symbol, the text from there to the next space character will be underlined but the space/s in between won't be (The Storm King). The other way is to tie the words together with a required space, the character, ^. For example, in &The^Storm^King the whole title spaces and all will be underlined as a unit (The Storm King). Although this is the preferred and easiest method, there can be a small price to pay in the .FI;AD format. If the title is towards the end of a line and cannot be fitted as a unit on that line, it will be wrapped in full onto the next line. That would leave the previous line with a great number of spaces between words just like the one about 11 or 12 lines up from here. You may find the appearance of such a sparse line not very elegant.

OVERSTRIKING with @

In a similar way to underlining in the paragraph above, the symbol @ is used to invoke overstriking. In doing this, the Text Formatter prints over the particular word four times before finally performing a line feed. The dark printing resulting stands out very clearly and is more

prominent than even the printer's emphasized style of print. If a group of words is required to be overstruck, the required space symbol (^) can be utilised in a similar fashion to that shown in the paragraph on underlining.

You must be wary not to use the &, @ or ^ characters for any other purpose although the & and the @ can be intentionally printed by typing the character twice. The first of the pair will be removed by the Formatter and only the second will be printed. The ^ will be removed and disappear from the printed copy.

TRANSLITERATES

This unusual word (Latin: litera=a letter and trans=across) simply means the use of one character to represent another character or even a group of characters. That's like assigning a # as a kind of variable to represent some long word or some printer code that might, say, change the print mode to enlarged italic elite print.

The formatter, when it encounters a circumflex (^) or an asterisk anywhere in the text or a dot (period) placed first on a line will remove them and execute some routine that is flagged by that symbol. To be able to purposely print one of those three, you need to transliterate some other little used characters to represent each of these three. That's the only way I was able to print the ^, and that was by TLing the accent grave to a circumflex. So, in the text, wherever I want a ^ I type an accent grave.

.TL 96:94 was placed on a line by itself near the top of the file to be printed. I know this sounds confusing but you have to actually use the process to follow what is happening. More on transliteration later.

INCLUDE FILE

This is another dot command that adds versatility to the Formatter. The command could look like .IF DSK1.RESUME2. When the Text Formatter encounters such a command, it prints the contents of that disk file too. The command can be at the beginning, the end or in the middle of the main file. There is no limit to the number of IFs used in the main file but IFs cannot be chained, i.e. a file that has itself been IFfed cannot have an IF command in it. Suppose you have done up a review that occupies 4 disk files named REVIEW1 to REVIEW4. Then to format them what you can do is to create another file to print out the whole review such as:

```
.CO Review:- The Storm King
.CO Date :- 23 March, 1991
.LM10;RM70 etc
.TL 96:42
.HE^^^^^^^^^^@THE^STORM^KING
.FO^^^^^^^^^^^^^^^^Page^%
.IF DSK1.REVIEW1
.IF DSK1.REVIEW2
.IF DSK1.REVIEW3
.IF DSK1.REVIEW4
```

A few more dot commands have cropped up here.

COMMENT

The .CO flags a comment just for the benefit of the reader. The whole line is ignored by the Formatter.

HEADER

.HE is the dot command that produces a heading on each page printed. The header is placed on the first line on the page and the text begins immediately on the next line. In the above example, the title, The Storm King, would be printed at the top of each page. The eleven circumflexes forces the title to be spaced over 11 positions from the indent position and the @ ensures that the heading is overstruck four times. Reusing the command, .HE, cancels any previous header command. If wording follows the new .HE, a new batch of headers can be printed otherwise the header position on the page is left blank.

FOOTER

.FO ensures that a footer is printed at the bottom of each page just after a blank line. The dot command can be followed by any text to be printed and/or the % character if required. The % instructs the Formatter to print consecutive page numbers. The example above will print both the word "Page" as well as its number. Page numbers can also be included in headers.

It seems that the F'web formatter prints out five lines less than the .PL command setting. On a setting of 60 lines per page, for example, the format printed is:

1 line either header or blank.
52 lines of text.
1 line blank.
1 line either footer or blank.

Concerning page numbers in headers or footers, what if we want to start numbering the pages from something other than one? Well, there's another command to take care of that.

PAGE NUMBER SET/RESET

.PA followed by a number sets header or footer pages to begin from that number. Relative values such as +2 or -1 can also be used to reset page numbers.

MORE ON TRANSLITERATION

I should make passing mention here of a valuable disk in the Club program library. Once you have got the general hang of transliterates and the IFfing of files ask John for Jack Shugrue's disk called PLUS!. Print out the documents and try out the files on the disk.

Here is a sample file based on Jack 's ideas that contains a lot of TLs for printer control through the Text Formatter. By using TLs most printer code sequences can be invoked by placing just a single character from ASCII 0 to 31 in the text. The file below has codes specifically for a Star printer and would suit most printers. The file should be saved as DSK1.*TL and stored on every disk you use for word processing. The reason for the asterisk in the filename is to ensure that this filename appears near the top of a directory listing and will not appear among the filenames of normal text files. To make use of the printer codes, one of the first lines of any text file should be .IF DSK1.*TL.

When typing the file below, firstly just type the transliterate code and press <ENTER> to get the ↵ symbols where they are shown. Then you can come back, in the non word wrap mode to type the comments after each.

```
.TL 0:32,27,64↵       @ Reset printer
.TL 1:32,27,77↵       A Elite print
.TL 2:27,72,32↵       B Dble strike off
.TL 3:32,15↵          C Condensed on
.TL 4:18,32↵          D Condensed off
.TL 5:32,27,69↵       E Emphasized on
.TL 6:27,70,32↵       F Emphasized off
.TL 7:32,27,71↵       G Dble strike on
.TL 8:8↵         H=Backspace reserved
.TL 9:32,27,52↵       I Italics on
.TL 10:10↵       J=Line feed reserved
.TL 11:27,53,32↵      K Italics off
.TL 12:12↵       L=Form feed reserved
.TL 13:13↵       M=Carr retn reserved
.TL 14:32,27,50↵      N 1/6 line spacin
.TL 15:32,27,48↵      O 1/8 line spacing
.TL 16:32,27,80↵      P Pica size print
.TL 17:32,27,51,17↵   Q sQuashed lines
.TL 18:32,27,83,0↵    R supeRscript
.TL 19:32,27,83,1↵    S Subscript
.TL 20:27,84,32↵      T Cancel sub/supe
.TL 21:32,27,45,1↵    U Underline on
.TL 22:27,45,0,32↵    V Underline off
.TL 23:32,27,87,1↵    W Wide enlarged on
.TL 24:27,87,0,32↵    X Wide enlarged o
.TL 25:32,27,120,1↵   Y NLQ characters
.TL 26:27,120,0,32↵   Z Draft chars
.TL 27:32,27,69,27,71,27,45,1↵ [ All on
.TL 28:42↵            \ Asterisk
.TL 29:27,70,27,72,27,45,0,32↵ ] All off
.TL 30:94↵            ^ Circumflex
.TL 31:46↵            _ Period
```

ALTERNATE CHARACTER SET

Before I can explain the contents of the TL file above, you will need to know about the alternate character set available in your WP. The normal set of characters range in ASCII values from 32 to 127 as shown in your Basic manual. The WP editor similarly limits the range of characters in the word wrap or non word wrap mode. But there is another mode provided in the editor where you can type ASCII values from 0 to 31. To toggle to and from this mode, you press CTRL/U. In this alternate input mode (but I will call it the CTRL/U mode from now on), the cursor appears as an underline and you can type characters from ASCII 0 to 31. Once in the CTRL/U mode keys of ASCII from 64 to 95 (@,A,B etc to a bit past Z) are reduced in value by 64 to make this ASCII range of 0 to 31. So pressing an "E" (ASCII 69) you get the character 5

(69-64=5).

With ASCII values below 32 you can directly control your printer as some of its codes include control numbers less than 32. These ASCII values are shown on the screen as tiny digits in hexadecimal. There are three exceptions though, being characters 10, 12 and 13 which appear as LF, FF and CR respectively. These produce special signals that control the paper feed and print head position of the printer. The file above uses transliteration of these characters overcoming the need for reserving any of those that are normally used in text. Looking at one of the TLs in the file:

```
.TL 5:32,27,69↵ E Emphasized on
```

5 is the ASCII character to use to start Emphasized printing. To type the 5, first press CTRL/U to go to the CTRL/U character mode with the underline type cursor, then press SHIFT/E (the fifth letter of the alphabet) and the tiny character 5 appears. Press CTRL/U again to get back to normal cursor mode. Type whatever word/s to be emphasized when printed then cancel the emphasis by typing the character, 6. That is, CTRL/U SHIFT/F and finally CTRL/U to return to normal cursor mode. Simpler when you're actually doing it rather than trying to grasp it mentally.

Referring again to the TL code above then, following the character 5 and the colon, are three ASCII values assigned to the character 5. If you consult your printer manual you'll find that 27 and 69 sent to the printer will enable the emphasized mode. 27 is called the ESCape character and 69 is "E", hence the sequence, ESC "E" shown in some printer manuals. Let's now consider the space character 32 that has been included with the 27 and 69. If the 5 were to be transliterated to the printer code for emphasized and encountered by the Formatter, the 5 would be removed from the text, acted upon in setting the printer code and the line filled to the right margin. But the Formatter fills to the right margin before removing the character 5, so that line will end up one character short of the right margin. The space character, therefore, is included to compenstate for that loss of one character.

If you intend to use the TL file above, you will need a copy to use as a reference when typing text. The main thing to record ia a list of the comments shown together with the mnemonic character for each. Then you can easily tell which SHIFT key to press while in CTRL/U mode to get the required printer response.

MAKING A SETUP FILE

Most times that you start up your word processor you need to set up your favourite tabs, margins and indent positions and on the first few lines to prepare a set of print margins, transliterates, comments etc. Why not have a standard layout on a special disk file that will do all of the above for you each time you need it? Then, when you start, its a simple matter to do a LoadFile of that filename and simply carry on typing using your own default screen margins, tabs and indent positions as well as the formatter attributes as outlined on those first few lines loaded. A sample of such a file with filename *SETUP could be:-

```
.LM10;RM70;IN+5;FI;AD;PL60↵
.IF DSK1.*TL↵
.CO Here place name/purpose of file↵
.HE If required↵
.FO If required↵
```

So each WP disk you use would have in readiness two files, *SETUP and *TL, on them.

Maybe you have been observant and noticed that, for all the times I've said in this series not to do this and not to do that, I have broken the rules I set. You've probably noticed some lines beginning with periods and several occurrences of asterisks and circumflexes etc. Transliteration has been the key to overcoming most of those hurdles which become limitations no longer. My only difficulty has been deciding whether I need to use an actual transliteration at a certain point or just to show an example of one as illustration. Anyway, the whole business of transliteration can be quite complicated if you like to go into it deeply as I found out when I set out to transliterate a tilde to download a ↵ symbol to my printer. The best way to learn all the intracies is firstly to have a need to use them and secondly to actually use the processes that produce the

results you require.

Next month there will be a fairly involved discussion (one-sided of course) on form letters and the use of mailing lists. To round the series off I will also include a list of hints and tips that come to mind and which may or may not have been mentioned in earlier articles. Following that I will include with a newsletter a loose sheet Reference Guide covering most of the commands/functions of the Word Processor.

oooooooooooooooooooooooooooooooooo

IN THE P.O. BOX

Micropendium, March 1992: Comments, Feedback, Learning to Print Letters (B), TI and the IBM (XB), Newsbytes, Off the End of the World (Ass), Reviews of Starbase Raiders, Classic Checkers, Grafics and Music, Payroll Files and Reports, Rattlesnake Bend, Zoom Flume, Castle Darkholm, and PrEditor, and User Notes.

TopIcs (LA99), July 1992: Ramblin' Thoughts from the President, Hole Currents #1, XB Miscelany, Managing your money, My Dear Aunt Sally.

TIt Bits (TI UP), December 1991: Editorial, Rescuing Ramdisks with a Corrupted ROS, Plus v2.0, Another Joystick Conversion, Printers, 'AT' Multifunction Card.

TIt Bits, March 1992: From the Editor, Disk Utilities, More on Printers, Archiving Revisited, Archiving - A Headache, XB to TIA Graphics, Beam Headings (XB program), Taking Control of your Formatter.

The Best Bits of TIt Bits: A compilation of articles from volumes 6, 7, and 8.

TIsHUG News Digest, July 1992: (RE - Memo on the cover. WHO SAYS !!!) Editor's Comment, Co-ordinator's Report, Secretary's Notebook, Shop, Software, Techo Time, TI World News, Funnelweb Editor, Review of the New Funnelweb Editor, Treasurer's Report, To See or not to C, TI-Bits, XB Tips, Letter to the Editor, Tigercub Programmable Calculator, Assembly Class, Multiplan Tips, Programming Tips and Reviews, TI-Base Tutorial, STRACC routine for Reformatter+, Beginning Forth, Writing in Machine Code.

WHAT'S NEWS

Garry Christensen

Bad news first. OPA still have not been able to get parts to complete the TIMs. They have been advised that we will be seeking a refund if there is no definate progress by the end of the month. They have also been told that although we may want a refund, what we really want is the cards. We may be able to come to some arrangement with regard to making orders COD. Discussions continue.

Better news. The order from Texaments has arrived. In fact it was probable waiting at the post office at the time of the last meeting but I was unable to clear the mail box at that time. The software will be available at the next meeting. Check the SHOP column for a listing.

I had a phone call from Bruce Campbell tonight. In the past week he has been busy harrassing the sysop of a BBS in Redcliffe to create a special area for the TI computer. His efforts have been successful. You can now up-load and down-load TI files and messages from that BBS. The membership is free. Phone numbers are 283 0314, 283 0315, 284 6853. Contact Bruce if you need more details or look for his article in either this or the next newsletter, depending if the editor gets the files in time.

Still no news on the order from Asgard. A letter has been sent so I hope we will see something soon.

It seems that Myarc are in the process of folding. The consumer affairs office in New Jersey advises that Lou Phillips is no longer required to repair any equipment but the faulty units must be returned to their owners. One of our members who's equipment had been returned to Myarc has been in contact with the consumer affairs office and has been advised that Myarc have dispatched some card(s) to me but as yet, nothing has arrived. At this stage we are assuming that they will be repaired.

All is not lost for owners of Myarc products. In fact, this could be good news. Cecure Electronics expect to become the authorized repair centre for Myarc products. They have been repairing cards for about 2 years and can fix all problems

ASGARD SOFTWARE HAS NOW ARRIVED

except those that involve the gate array, PAL chips or circuit boards. They are presently negotiating for access to the details of these parts. When complete, Cecure Electronics hope to be able to manufacture both Geneve and HFDC as well as repair them and they are looking at releasing them in kit form and to supply parts.

As for the software, Beery Miller reports that donations have so far raised 65% of the funds necessary to purchase MDOS. It is expected that Beery will be coordinating future software efforts. I beleive that the removal of Myarc from the scene will result in the opening up of both the hardware and software architecture and the end result will be more information and better software, as well as more efficient hardware production and support.

ESD, who are developing a new hard disk controller, did not demonstrate the device at the Lima Conference in May. Although it was scheduled for release in April, the rumours are that is will not be ready till the latter part of the year.

Texaments are offering package deals for software. #1 - Gif Mania, TIA, and Sound F/X for $49.95. #2 - Sound F/X and 6 disks of sound files for $21.95. #3 - TIA, Gif Mania and 2 Artist Companions for $39.95. Prices in US dollars and do not include postage. Texaments, 53 Center St, Parchogue NY 11772, USA.

Lima User Group are offering 3 video tapes covering 17 hours of their recent conference. They are available for $15. The list of speakers reads like a who's who of the TI world and cover just about everything. Contact Charles Good, PO Box 647, Venedocia OH 45894, USA.

The beta test of the new Funnelweb editor is out and from what I have read, it will be great. I look forward to getting my hands on the release version when the time comes.

Finally, Fred Cugley in Adelaide has a stand-alone DSDD disk controller and a speech synthasizer for sale. He is open to offers. He also has a lot of other stuff that he hasn't had time to catalogue yet. You can phone him on (08) 258 3499.

OOPS

Last month's newsletter included a listing for the TigerCub Reformatter. Unfortunately, this program includes an embedded assembly program and will not run if you type it in as listed. Please contact Garry if you want a copy of the program.

oooooooooooooooooooooooooooooooooo

SHOP

Just a few words about the new softeware that has come from Texaments.

The Organizer and Publications Index are applications for TI-Base so they will only benefit you if you have that program. TI-Base is readily acknowledged as one of the best database programs for the TI however you will probably have to have some programming experience to make full use of it. These two programs are applications that do not require you to have any special programming knowledge as they are already set up.

The Organizer is for general purpose storage of information. It is structured like a filing cabinet. There are 4 draws that you can store information in and each draw is divided into folders. There can be up to 16,129 folders in each draw, assuming you have that much space on you disk. Each folder is given a name and can store 6 lines of text. You can use this text to store any information that you like. Accessing the information is as easy as opening the draw and then opening the folder.

Publications Index is a little more explicit. It is used for cataloguing all those articles in magazines or periodicals. For each article you store the name, the name of the magazine, the type of article, the date and page number for the magazine, the author, the key category and an indexing character. The program allows creation of subfiles.

TI Sort is also commonly used with TI-Base but it is not restricted to that program. TI-Sort will sort records in many different types of files. It is a lot faster than the sort utility in TI-Base

The Missing Link has been demonstrated in the group before. It is a set of utilities for Extended Basic that provide quite startling results. If you program in XB, this program is certainly worth the look. There are also quite a number of programs around now that will only work with TML. There is a demonstration disk in the library.

Rapid Copy has been around for a long time. This program allows you to copy disks very quickly. From memory, it takes about 100 seconds to copy a DSDD disk, including formatting the target disk. If you have a copy but haven't paid for it (you know if I am talking about you), here is a great chance to make it legal. If you haven't got a copy, its really worth it.

The last 2 are for use with TI-Artist. Artoons is a set of 3 disks full of your favourite cartoon characters in instance format. Use them to decorate TIA pictures or with any other program that will make use of instances. The other program is display master. This program allows you to setup a slide show of your TI-Artist pictures.

All the software listed below is in stock. Prices are in Australian dollars and include postage from the USA. Please add the cost of postage within Australia, if applicable.

Rock Runner	$14.00
Waterworks	$14.00
Beyond Video Chess	$11.50
Rattlesnake Bend	$ 7.75
Castle Darkholm	$ 9.75
Doom Games I	$ 7.75
Doom Games II	$ 7.75
Doom Games III	$ 7.75
Page Pro	$27.00
Page Pro Pics #1	$ 8.50
Page Pro Pics #2	$ 8.50
Page Pro Pics #3	$ 8.50
Page Pro Pics #4	$ 8.50
Page Pro Pics #5	$ 8.50
Page Pro Pics #6	$ 8.50
Page Pro Pics #10	$ 8.50
Page Pro Pics #14	$ 8.50
Page Pro Pics #15	$ 8.50
Page Pro Borders #1	$ 8.50
Page Pro Borders #2	$ 8.50
Page Pro Fonts #1	$ 8.50
Page Pro Fonts #2	$ 8.50
Page Pro FX	$16.50
Page Pro Headline Maker	$11.50
Page Pro Headline Fonts #1	$ 8.50
Page Pro Headline Fonts #2	$ 8.50
Page Pro Templates #5	$ 7.70
Page Pro Templates #6	$ 7.50
Page Pro Templates #7	$ 7.50
Pix Pro	$16.00
Quick Run	$11.00
Microsoft Multiplan	$20.00
LOGO II	$20.00
The Missing Link	$35.00
Display Master	$25.00
Artoons	$20.00
Publications Index	$25.00
The Organizer	$25.00
TI Sort	$25.00
Rapid Copy	$22.00

oooooooooooooooooooooooooooooo

LAST MEETING

Meeting opened at 8:35 pm.

Minutes read and accepted. Moved John Reynolds, Sec John Campbell.

Business Arising: OPA still waiting for crystals for TIMs, Col advised that some of the Cadet's were ready.

Correspondance : as per last newsletter with letters to Jim Peterson and Asgard outgoing. Accepted, Moved Bruce Campbell, Seconded John Campbell.

Treasurer's Report: $331.80 in bank, $482 in cash received.

General Business:

John Peacock reminded members who have modules on loan to return them.

Col moved, Sec John Peacock, the Chas be paid $264 for the speech cards. Carried.

Bruce Campbell moved, sec Col, that he be refunded the $19.80 for the transformer for the MBX. Carried.

Larry offered to demonstrate the Text To Speech software at the next meeting. John Reynolds moved, sec Rex, that we purchase another 200 disks. Carried.

Garry tendered his resignation for the position of Editor. John Peacock offered to take over the position.

An order for Tex-Comp is being considered. Please place orders for any software that you may want.

Meeting closed at 9:30pm

Col gave a demonstration of the Cadet.

TIPS FROM THE TIGERCUB

41

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

The answer to the challenge in the last Tips? For a clue, try -
DISPLAY AT(24,1):0 in Basic.

Still don't get it? In Basic, DISPLAY is the same as PRINT, but AT is not recognized, so the computer thinks you are telling it to print the variable AT(1,1) - which, being undefined, is 0 - and advance to the next line (the :) and print 0.

I have always wanted a pocket calculator with several memories and a window to display the contents of each one. So, since there is plenty of room for windows on a TV screen, I wrote one.

It does not require any use of the Enter key, but each CALL KEY input must be validated and processed, so don't type too fast. It will accept such inputs as M1=7= or M1=7+1= or M2=1-M1= to put a value in a memory, or 6+7= or 6+M2= to calculate and display, or 6+7M1 or M1-.M2M3 to calculate and put into memory, and will even do multiple calculations such as 1+2-3/4*5%6, subtotaling after the first two.

```
100 CALL CLEAR :: CALL SCREEN(5):: DEF S$(X)=SEG$(A$,X,1)&" = " :: CALL PEEK(8198,A) :: IF A<>170 THEN CALL INIT
110 CALL LOAD(-31806,16):: ON WARNING NEXT :: GOTO 140
120 SET,M$(),K,S,A$,S$(),R,C,N,N1,N2,N1F,N2F,M1F,M,MF,DF,FF,VF,EF,FL,N$,F2,T,M2,MEM(),ST,NX,ZF
130 CALL COLOR :: CALL CHAR :: CALL KEY :: CALL SOUND !@P-
140 FOR SET=0 TO 4 :: CALL COLOR(SET,16,1):: NEXT SET :: FOR SET=5 TO 8 :: CALL COLOR(SET,5,16):: NEXT SET :: CALL CHAR(64,"0")
150 FOR SET=9 TO 12 :: CALL COLOR(SET,16,1):: NEXT SET
160 DISPLAY AT(1,10):"TIGERCUB":"  MULTIMEMORY@CALCULATOR": :"MEMORY #1": :"MEMORY #2": :"MEMORY #3": :"MEMORY #4": :"MEMORY #5"
170 M$(1)="0123456789.+-*/%=CXM" :: M$(2)="0123456789.ASMDPECXM" :: DISPLAY AT(20,1):"use ?":"(1) symbols":"(2) alpha characters"
180 CALL KEY(0,K,S):: IF S=0 OR K<49 OR K>50 THEN 180 :: A$=M$(K-48)
190 DISPLAY AT(20,1):S$(12);"add";TAB(16);S$(16);"percent" :: DISPLAY AT(21,1):S$(13);"subtract";TAB(16);S$(17);"equals"
200 DISPLAY AT(22,1):S$(14);"multiply";TAB(16);S$(18);"cancel" :: DISPLAY AT(23,1):S$(15);"divide by";TAB(16);S$(19);"clear all"
210 DISPLAY AT(24,1):"M1 to M5 = memories #1 to #5"
220 R=15 :: C=1 :: N,N1,N2,N1F,N2F,M1F,M,MF,DF,FF,VF,EF,FL,ZF=0 :: N$="" :: DISPLAY AT(18,1):""
230 CALL KEY(3,K,S):: IF S<1 THEN 230 :: CALL SOUND(50,500,5):: DISPLAY AT(R,C):CHR$(K):: C=C+1
240 ON POS(A$,CHR$(K),1)+1 GOTO 260,270,270,270,270,270,270,270,270,270,280,290,250,290,290,290,340,410,420,430
250 IF VF=1 OR MF=1 THEN 290 :: ZF=1 :: N$="-" :: GOTO 230
260 DISPLAY AT(R,C-1):"?" :: C=C-1 :: GOTO 230
270 IF MF=1 THEN 260 :: FL=0 :: VF=1 :: IF DF=0 AND ZF=0 THEN N=N*10+K-48 :: GOTO 230 ELSE N$=N$&CHR$(K):: GOTO 230
280 IF DF=1 THEN 260 :: DF=1 :: MF,FL=0 :: IF ZF=1 THEN N$=N$&"." :: GOTO 230 ELSE N$=STR$(N)&"." :: GOTO 230
290 IF C=2 OR FL=1 THEN 260 :: FL=1 :: IF FF=0 THEN 320
300 F2=POS(A$,CHR$(K),1)-11 :: IF VF=1 THEN GOSUB 480
310 GOSUB 520 :: N1=T :: DISPLAY AT(18,1):"SUBTOTAL";T :: N2F,N2=0 :: FF=F2 :: GOTO 230
320 IF VF=0 THEN 330 :: VF,MF=0 :: GOSUB 480
330 MF=0 :: FF=POS(A$,CHR$(K),1)-11 :: GOTO 230
340 IF C=2 OR(FF=0 AND M1F=0)OR(C=4 AND M1F=0)OR FL=1 THEN 260
350 IF C=4 THEN EF=1 :: M2=M :: N1F,MF=0 :: GOTO 230
360 IF VF=1 THEN GOSUB 480
370 IF EF=0 THEN 400
380 IF N2F=0 THEN MEM(M2)=N1 :: DISPLAY AT(M2*2+2,11):N1 :: GOTO 220
390 GOSUB 520 :: MEM(M2)=T :: DISPLAY AT(M2*2+2,11):T :: GOTO 220
400 GOSUB 520 :: DISPLAY AT(15,C):T :: GOTO 220
410 DISPLAY AT(R,1):"":"":"":"" :: GOTO 220
420 MEM(1),MEM(2),MEM(3),MEM(4),MEM(5)=0 :: FOR R=4 TO 12 STEP 2 :: DISPLAY AT(R,10):"" :: NEXT R :: GOTO 410
430 IF EF=1 AND MF=1 THEN 260
440 CALL KEY(3,K,ST):: IF ST<1 OR K<49 OR K>53 THEN 430 ELSE CALL SOUND(50,500,5):: M=K-48 :: DISPLAY AT(R,C):CHR$(K);:: C=C+1 :: MF=1 :: FL=0 :: IF VF=1 THEN GOSUB 480
450 IF N1F=0 THEN M1F,N1F=1 :: N1=MEM(M):: IF ZF=1 OR DF=1 THEN N1=VAL(N$&STR$(N1)):: DF,ZF=0 :: GOTO 230 ELSE 230
460 IF N2F=0 THEN N2F=1 :: N2=MEM(M):: IF ZF=1 OR DF=1 THEN N2=VAL(N$&STR$(N2)):: DF,ZF=0 :: GOTO 230 ELSE 230
470 GOSUB 520 :: MEM(M)=T :: DISPLAY AT(M*2+2,11):T :: GOTO 220
```

```
480 IF DF=0 AND ZF=0 THEN NX
=N ELSE NX=VAL(N$):: DF,ZF=0
490 IF N1F=0 THEN N1=NX :: N
1F=1 :: GOTO 510
500 N2=NX :: N2F=1
510 VF,N=0 :: N$="" :: RETUR
N
520 IF FF=1 THEN T=N1+N2 ELS
E IF FF=2 THEN T=N1-N2 ELSE
IF FF=3 THEN T=N1*N2 ELSE IF
 FF=4 THEN T=N1/N2 ELSE T=N1
*N2/100
530 RETURN
```

I have always been annoyed by the difficulty of hyphenating with TI-Writer, when I want to avoid the gaping holes that wraparound and Fill and Adjust can cause. Manually filling and adjusting with carets is slow, and leaving a space after the hyphen is unreliable, so I wrote this program.
(see improvement in Tips 42 - Ed.)

```
100 DISPLAY AT(2,10)ERASE AL
L:"TIGERCUB":"  HYPHENATED F
ILL AND ADJUST"
110 DISPLAY AT(6,1):" Prepar
e text with TI-Writer":"Edit
or. Leave left TAB at 0,":"s
et right TAB  at the actual"
:"value of the line length d
e-"
120 DISPLAY AT(10,1):"sired
(i.e.,  for  a 28-char":"lin
e, set it at 28)."
130 DISPLAY AT(12,1):" Inden
t  as  desired. Center":"hea
dings  as  desired but be":"
sure  to  follow them with a
":"line feed (Enter). Hyphen
ate"
140 DISPLAY AT(16,1):"as  de
sired  and  follow the":"hyp
hen  immediately  with  a":"
line feed (Enter)."
150 ON ERROR 160 :: GOTO 170
160 ON ERROR 160 :: RETURN 1
70
170 DISPLAY AT(20,1):"INPUT
FILE? DSK" :: ACCEPT AT(20,1
6)BEEP:F$ :: OPEN #1:"DSK"&F
$,INPUT
180 DISPLAY AT(22,1):"OUTPUT
 FILE? DSK" :: ACCEPT AT(22,
17)BEEP:NF$ :: OPEN #2:"DSK"
&NF$,OUTPUT
190 DISPLAY AT(24,1):"LINE L
ENGTH?" :: ACCEPT AT(24,14)V
ALIDATE(DIGIT):L
200 LF$=CHR$(13):: H$="-"&CH
R$(13)
210 ON ERROR 210 :: GOTO 220
220 ON ERROR 210 :: RETURN 3
10
230 LINPUT #1:M$ :: IF M$="
" OR M$=LF$ OR M$="" OR ASC(
M$)>127 OR(LEN(M$)=L AND POS
(M$,LF$,1)=0)OR POS(M$," ",1
)=0 THEN 310
240 IF POS(M$,LF$,1)<>0 AND
POS(M$,H$,1)=0 THEN 310
250 IF POS(M$,H$,1)<>0 THEN
M$=SEG$(M$,1,LEN(M$)-1)
260 IF LEN(M$)=L THEN 310
270 P=1
280 X=POS(M$," ",P):: IF X=P
 THEN P=P+1 :: GOTO 280 ELSE
 Y,P=X :: IF POS(M$," ",P)=0
 OR P=L THEN 310
290 M$=SEG$(M$,1,X)&" "&SEG$
(M$,X+1,255):: IF LEN(M$)>=L
 THEN 310 ELSE P=X+2
300 X=POS(M$," ",P):: IF X=0
 THEN P=Y :: GOTO 300 ELSE G
OTO 290
310 PRINT #2:M$ :: IF EOF(1)
<>1 THEN 230 ELSE CLOSE #1 :
: CLOSE #2
```

Here is one for the preschoolers -

```
100 CALL CLEAR :: CALL SCREE
N(14):: CALL COLOR(1,11,11,1
2,5,5):: DISPLAY AT(3,10):"S
EE-N-SAY": : :"PRESS ANY KEY
" !by Jim Peterson based on
a routine by Michael Lyons
110 DIM E$(16),PAT$(16):: CA
LL CHAR(123,RPT$("F",16))
120 DATA "    ","   {","  {
","  {{","{  ","  { {","  {{
","  {{{","{   ","{  {","  {{{
","{{ {","{{  ","{{{ {","{{{
","{{{{"
130 FOR J=0 TO 15 :: READ PA
T$(J):: NEXT J
140 CALL KEY(0,K,S):: IF S=0
 THEN 140
150 CALL CHARPAT(K,CP$):: FO
R X=1 TO 16 :: Y=ASC(SEG$(CP
$,X,1)):: E$(X)=PAT$(Y+(Y>57
)*7-48):: NEXT X :: IF K>96
AND K<123 THEN K=K-32
160 CALL CLEAR :: CALL SAY(C
HR$(K)):: FOR X=2 TO 16 STEP
 2 :: DISPLAY AT(8+(X/2),12)
:E$(X-1);E$(X):: NEXT X
170 CALL SAY(CHR$(K)):: GOTO
 140
```

And so, one more time

MEMORY FULL

Jim Peterson

Courtesy of TIsHUG

Sorting part 5

by Ron Brubaker, USA

An introduction to the shell sort was the subject of last month's article. This article is intended to show that the shell sort is amendable to all of the variations shown previously for the bubble sort. First of all, consider the following program which utilises the same sorting routine shown at the end of last month's article but to sort a different set of data.

```
10 REM *** READ IN A LIST OF RANDOMLY ORDERED NAMES ***
20 REM
30 DIM A$(15),R(15),P(15)
40 READ N
50 FOR I=1 TO N
60 P(I)=I
70 READ A$(I),R(I)
80 PRINT R(I);
90 NEXT I
100 DATA 15
110 DATA "WASHINGTON, GEORGE",1,"JEFFERSON, THOMAS",3
120 DATA "FORD, GERALD",37,"KENNEDY, JOHN",34
130 DATA "FILLMORE, MILLARD",13,"ARTHUR, CHESTER",21
140 DATA "ADAMS, JOHN Q.",6,"LINCOLN, ABRAHAM",16
150 DATA "ROOSEVELT, FRANKLIN",31,"REAGAN, RONALD",39
160 DATA "CARTER, JAMES",38,"WILSON, WOODROW",27
170 DATA "MONROE, JAMES",5,"ROOSEVELT, THEODORE",25
180 DATA "ADAMS, JOHN",2
190 REM
200 REM ***** SHELL SORT ROUTINE - NUMERIC *****
210 REM
220 L=(2*INT(LOG(N)/LOG(2)))-1
230 L=INT(L/2)
240 IF L<1 THEN 390
250 FOR J=1 TO L
260 FOR I=J+L TO N STEP L
270 K=I
280 T=R(I)
290 IF R(I-L)<=T THEN 330
300 R(I)=R(I-L)
310 I=I-L
320 IF I>L THEN 290
330 R(I)=T
340 I=K
350 NEXT I
360 NEXT J
370 GOTO 230
380 REM
390 REM *** ROUTINE TO PRINT SORTED LIST OF NUMBERS ***
400 REM
410 PRINT
420 FOR I=1 TO N
430 PRINT R(I);
440 NEXT I
450 END
```

In this form the program will sort the list of presidents in the order in which they held office. If an alphabetic sort is preferred simply change the middle section as follows:

```
10 REM *** READ IN A LIST OF RANDOMLY ORDERED NAMES ***
20 REM
30 DIM A$(15),R(15),P(15)
40 READ N
50 FOR I=1 TO N
60 P(I)=I
70 READ A$(I),R(I)
80 PRINT R(I);
90 NEXT I
100 DATA 15
110 DATA "EDMUND, BARTON      ",1,"WATSON, JOHN       ",3
120 DATA "HOLT, HAROLD        ",17,"HUGHES, WILLIAM     ",7
130 DATA "CHIFLEY, BEN        ",16,"DEAKIN, ALFRED      ",2
140 DATA "FRASER, MALCOLM     ",22,"MENZIES, ROBERT     ",12
150 DATA "GORTON, JOHN        ",19,"WHITLAM, GOUGH      ",21
160 DATA "REID, GEORGE        ",4,"CURTIN, JOHN       ",14
170 DATA "McEWEN, JOHN        ",18,"SCULLIN, JAMES      ",9
180 DATA "McMAHON, WILLIAM",20
190 REM
200 REM ***** SHELL SORT ROUTINE - ALPHABETIC *****
210 REM
220 L=(2*INT(LOG(N)/LOG(2)))-1
230 L=INT(L/2)
240 IF L<1 THEN 390
250 FOR J=1 TO L
260 FOR I=J+L TO N STEP L
270 K=I
280 T$=A$(I)
290 IF A$(I-L)<=T$ THEN 330
300 A$(I)=A$(I-L)
310 I=I-L
320 IF I>L THEN 290
330 A$(I)=T$
340 I=K
350 NEXT I
360 NEXT J
370 GOTO 230
380 REM
390 REM *** ROUTINE TO PRINT SORTED LIST OF DATA ***
400 REM
410 PRINT
420 FOR I=1 TO N
430 PRINT A$(I);
440 NEXT I
450 END
```

As you will notice from running this program, the list of Australian Prime Ministers will be sorted in alphabetic order. As a school teacher it is interesting to note that there is always a student in every class that knows America's first president (George Washington, who live 200 years ago), but it is rare to find a student who knows Australia's first Prime Minister (Edmund Barton, who lived less than 100 years ago!). Sometimes I wonder where our priorities are!

If you prefer to use a pointer sort then it is possible to write a short program for numeric and string variables. Perhaps this could be done as an exercise?

Next month will probably be the last in this series. It will introduce a sorting algorithm called Quicksort II which is one of the fastest sorting methods. o

3.5 inch DISK DRIVES

Geoff WARNER

Being a member of TIsHUG, our counterpart in Sydney, has a lot of advantages, not the least being the opportunity to take advantage of the many special deals that they seem to be able to negotiate on software AND hardware

I took advantage of a recent offer on SSDD 3.5" disk drives at $20.00 per unit. One of the reasons was the relative indestructability of the disk itself, as I had experienced some problems with 5.25" di sks jamming in their jackets of late. One was a disk of unknown pedigree that I used on my TI - 99 / 4A (AND lost my TI BASE data for the girls' basketball team as I was - foolishly - using the same disk for BOTH application AND data storage) and the other was a programme disk from a relatively well - known source of IBM and compatible software that I am forced to use in my occupation. I felt that the rigid case of the 3.5" disk would save me more headaches of the floppy disk variety

The fact that the 3.5" drive was of the SSDD variety didn't bother me as (i) I have purchased a TIsHUG 'AT' MULTIFUNCTION CARD and DSDD disk controller kit (more on that later) so I could see the DD option giving me the same capacity as a DSSD drive for only $20.00 - surely a bargain

(ii)$20.00 for a new drive is the deal of a lifetime anyway

I am pleased to report that the drive functioned perfectly FIRST TIME after I had followed these steps :

- connect the drive using the appropriate connector (the same type as one on the 'internal' connector on the disk controller card

- set the drive number using the supplied connector (remembering that the first drive is often referred to as DRIVE 0)

- remove the termination, or shunt, pack from the last drive connected to your system and set the new drive as the last drive in your system

As 3.5" disks are becoming less expensive almost daily, (currently < $1.00 each in Perth) I would recommend the addition of a 3.5" drive to your system. The addition of a CORCOMP, MYARC or TIsHUG disk controller card will double your existing disk capacity

WHY NOT GO FOR IT ? The speed and quietness of operation are well worth the expense (even at DSDD drive prices, approximately $95.00 in Perth), which incidentally, is about the same as a 5.25" drive of lesser storage capacity. The good part is that most IBM - compatible drives will work with the TI and that, due to the current market situation, most dealers are VERY co - operative when it comes to making their items function in the manner intended.

Courtesy of TIsHUG

G, an Introduction

by Bob Warren, ATICC

G can stand for many things, including GRAPHICS, GREAT, and GOOD FUN. The graphics language G is all of these things. It was developed by Gene Krawczyk, one of the original members of ATICC (in Adelaide, ed.), and he has made it available to the Club. It was written in Assembly Language, and so requires a disk drive and memory expansion to run.

G is a powerful graphics language with simple commands, which makes it ideal for children, but it is also sophisticated enough to provide a challenge for adults. Unlike TI-Artist, which only produces still pictures, G allows a form of animation, and can rapidly change screens. A screen can be STOREd at any time, a new screen drawn, and then the old screen RESTOREd when required. A screen can also be SAVEd to, or LOADed from, a disk when needed.

I have edited Gene's instructions and commands to produce a beginner's version of G which everyone can use. Details of this version are given below. Gene also included some commands which allow an experienced Assembly Language programmer to execute Assembly routines, including sprites (I think). When I have sorted these out and written instructions for their use, I will present an Advanced G (not available at this time, ed.).

I hope there will be sufficient interest for a regular column in our bulletin to be devoted to G, both to answer queries and to exchange programming ideas. o

G, the HIRES Language

by Gene Krawczyk, ATICC
Documentation by Bob Warren, ATICC

Loading and operating G

To load G from Extended BASIC insert the G system disk into drive #1 and select Extended BASIC. This action automatically loads G into memory, and the program will then attempt to load & run a file called 'GLOAD', if this file is present on the disk (in the same way that the system attempts to load & run a file called 'LOAD' everytime you select Extended BASIC). In this instance 'GLOAD' is a demonstration program, but any program can be called 'GLOAD', if you want it to automatically run everytime to boot up G. If 'GLOAD' is not on the disk in drive #1, the program will immediately go into menu mode, described below.

To load G with the Editor/Assembler module, select '5' from the E/A menu, then type in 'DSK1.GEE' (without the quotes) and press <enter>. The program will boot up by running the 'GLOAD' file descibed previously.

To escape at any time press FCTN <9>, which will return you to the menu mode. The screen displays the following message:

G a graphics programming language
(L)oad,(S)ave,(I)nit,(E)dit,(R)un.

Pressing any of these five keys will put you into that mode.

(L)oad
Expects input from the keyboard, e.g. DSK1.SAMPLE will load the file 'SAMPLE' into memory. These files are in DIS/VAR 80 format without any control characters such as carriage returns. If you load another file the old file will be lost. If the load is successful, the program will be tokenised and then run.

(S)ave
Will save the file in memory, e.g. DSK1.SAMPLE will save what ever is in memory to the disk drive. If there is nothing in memory then it will save an empty file to the disk.. Note: loading an empty file into memory (e.g. 'DSK1.NEW') is a good way to clear the memory, since no command exists which can do this directly. You can also save to a printer or other pheripheral by typing in the required device name, e.g. TP will save the file to the thermal printer.

(I)nit
If you select (Y)es to the prompt to initialise a file, the program will clear the memory pointers and restart. This is similiar to a cold start and the program will try to auto boot a file called 'GLOAD' from drive #1.

(E)dit
Selecting 'E' will put you into the Editor. The key presses used to move the cursor around the screen are the same as the ones used in TI Writer. The main difference is that this editor does not window its screen. Instead, each line of 80 columns is displayed as two consecutive rows of forty columns.

The following key presses are used in the Editor:

To delete a line press FCTN <3>
To insert a line press FCTN <8>
To scroll the screen up press FCTN <4>
To scroll the screen down press FCTN <6>
To delete a character press FCTN <1>
To insert a character press FCTN <2>
To go back to the menu press FCTN <9>
To run the file in memory press FCTN <5>

(R)un
This command does the same as in TI BASIC.

Now for a few simple programs to get your feet wet. Note: anything in parentheses is the key press you have to make. From the menu mode, with the original disk in drive #1 press (L) to load a file. Type in: DSK1.GLOAD <enter>. This will load and automatically run a small demonstration. To stop the demonstration, press and hold FCTN <9>; this will take you back to the menu. To look at the program, press (E) to enter the editor.

All commands will be typed in uppercase on the left side of the screen. A small example will be shown first and then explanations.

```
SET 10 20 TO 100 150
```

This will draw a line starting at location 10 across and 20 down as the starting point. The line will then be plotted to position 100 across and 150 down. All co-ordinates are given as x y, the x is columns and the y is rows. The value of x can be from 0 to 255, and the value of y is 0 to 191. Notice also there are no commas or parentheses in this language to separate variables or commands. The separator is a space (the same as in FORTH).

```
SET 10 24 TO 134 180
STOP
```

This will put a line on the screen and then it will just stop. When you press a key you will be taken back to the command mode. The actual command SET puts a dot on the screen and the command TO draws the line from the old position to the new co-ordinate. This allows you to have as many TOs as you like to draw shapes. So the command TO always draws from the last plotted point to the new one. If you just want to put a dot on the screen you would use:

```
SET 100 100
```

Description of Commands

ANGLE
Sets a new angle for the DRAW command and can be between 0 to 360 degrees. Other values can be used,

but if numbers greater than 360 are used, they are just reduced to a value between 0-360. The use of larger numbers may also slow down execution time. For example:

```
ANGLE 250
```

sets the angle to 250 degrees.

ARC
Allows you to draw circles and arcs, e.g.

```
ARC 100 100 25 25 0 360
```

draws a circle at co-ordinates x=100, y=100 with a diameter of 25 pixels.

ARRAYS
Only one dimensional arrays are allowed, in the format @(a), where 'a' can be any integer between 0 and 500, expressed as a decimal or hexadecimal number.

BCOLOR c

Same as COLOR, but changes the background color. The default is clear. Seems only to be used in TRACE 4.

BGND
Sets the background color. The same rules apply as for FGND, but only works on the background colors.

BOX x y a b
Draws a box on the screen whose top left hand corner has the co-ordinates of 'x' and 'y'. The box is 'a' amount wide and 'b' amount tall, e.g.

```
BOX 100 100 50 20
```

puts a box on the screen which is 50 pixels wide and 20 pixels tall, with the top left hand corner located at x=100 and y=100, remembering that the left hand corner of the screen has the co-ordinates x=0, y=0.

CLEAR
Clears an area of the screen.

CLS
Clears the graphic screen without first saving it. If you accidentally use this command, you would have to design this screen again.

COLOR c
Changes the foreground pen color to a new value. The value is from 0 to 15. Any value larger than this is truncated.

CONSTANTS
Constants can be integers, in decimal or hexadecimal format.

COS
Gives the cosine of a number.

DISPLAY
Prints data to the screen, but erases what ever is underneath.

DRAW
Draws a line from the last point plotted, at the angle specified in the ANGLE command.

FGND
Sets the foreground color. For example,

FGND 6 sets the foreground color of all "on" pixels to red.
FGND 6 0 255 sets the "on" pixels from 0 to 255 to red.

FILL x y
Fills in a area on the screen. This is not a very exact type of fill but it may do the job. It might have to be used a few times, however, to fill in awkward shapes. You may have problems with this command.

FOR NEXT STEP (optional)

Same as in TI BASIC, but only 10 levels of nesting are allowed, e.g.

```
FOR I=1 TO 1000 NEXT I
```

produces a delay.

FORMAT
Sets up the distance between characters for use in the print statement. The normal distance is 8, and in forty column mode the normal distance is 6.

GOSUB :LABEL
Same as GOTO except a return is stored so you can do subroutines. There are only forty levels of subroutines allowed in G.

GOTO :LABEL
Does a branch to the label in the file, e.g.

```
:SAM GOTO :SAM
```

would give the same result as 100 GOTO 100 in TI BASIC.

All labels must be preceded by a colon (:), and must not be more than six characters in length, seven counting the colon. Also the label must be the first word on a line, otherwise the search routine will not find your label. If you put a space in front of the label the word will also be missed.

IF THEN
Similiar to the TI BASIC 'IF' expression. If true then do rest of line, else go to next line. You can not do this though:

```
IF A=10 THEN :LABEL
```

is not correct; you must do this instead:

```
IF A=10 THEN GOTO :LABEL
```

INVERT
Inverts an area of the screen.

KEY$
KEY$ A puts the key press into variable A. No key press means A=255.

LET
Assigns a value to a variable, e.g.

```
LET A=10
```

gives 'A' the value of 10, and

```
LET A=10*3+4/2+C*B
```

will give 'A' the value of the calculation. The value is worked out from left to right without precedence among the mathematical operators, as there is in TI BASIC. Also you cannot use parentheses to change the way you want the program to calculate the value.

As in TI BASIC the LET comand is optional, but if you want speed use it, as LET quickens the processing time. Without LET there is a lot of checking.

LINESTYLE
Changes the type of line being used.

LOADS
Loads a bit map screen from disk in GRAPHX format, e.g.

```
LOADS DSK1.SAMPLE
```

loads a screen called 'SAMPLE' from drive #1.

PATTERN
Alters the fill pattern.

uses whole numbers, SIN and COS would only return the numbers -1, 0 or +1, which would not be of much use. In most programs sin and cos are usually multiplied by some other number anyway, so Gene set up the commands as: SIN A B, and COS C D, where A and C are the angles, B and D are the numbers multiplying sin and cos. B and D are also the variables to which the values are returned. For instance,

```
LET B=10 SIN 30 B
```

sets B equal to 5, since sin 30 = 0.5, so 0.5=5.

Try this little program, it is as easy as ABC.

```
A=128 B=96
V=90 W=60 F=1 G=3
SCREEN 1
C=6
FGND C
FOR I=1 TO 5
A=A+1 C=C+1
FOR T=0 TO 360
LET M=V LET N=W
COS F*T M
SIN G*T N
SET A+M B+N
NEXT T
FGND C
NEXT I
STOP
```

Also try other values for the variables F and G and see what happens. In this program SIN and COS are immediately multiplied by the pixel locations on the screen, and so the results are always large whole numbers. In other situations where you may wish to use SIN and COS you may have to multiply them by a large number, say 100, do the calculation, and then divide by 100 to get the answer to the required accuracy.

You can use the same technique of using a variable in different functions to plot all sorts of patterns.

As an example of simple animation, try this little program.

```
REM MOVING HELLO
FOR I=8 TO 128 STEP 4
DISPLAY I 20 " HELLO"
REM NOTE THE SPACE
NEXT I
STOP
```

If anyone has any problems or suggestions (or even reads this column), I would be glad to hear from them.

o

SIXTEEN - Extended Basic Program

```
100 !!!!!!!!!!!!!!!!!!!!!!!!
110 !    SIXTEEN PUZZLE    !
120 !      AUGUST 83       !
130 !!!!!!!!!!!!!!!!!!!!!!!!
140 GOSUB 200 ! INITIALIZE
150 GOSUB 1000 ! SETUP
160 GOSUB 1200 ! MOVES
170 GOSUB 2000 ! RESULT
180 GOTO 160
190 END
197 !!!!!!!!!!!!!!!!!!!!!!!!
198 ! INITIALIZE           !
199 !!!!!!!!!!!!!!!!!!!!!!!!
200 DISPLAY AT(10,8)ERASE AL
L:"SIXTEEN PUZZLE": : :TAB(9
);"D. WHITE 1983"
210 OPTION BASE 1 :: DIM L$(
16),O(16),M(4,16),R(16),C(16
),P(16)
220 CALL COLOR(9,2,9,10,2,9,
11,2,9,12,2,9)
230 CALL CHAR(96,"FF80808080
808080FF0101010101010180808080
80808080FF01010101010101FF")
240 CALL CHAR(100,"FF0000000
0000000808080808080808001010
1010101010100000000000000FF"
)
250 CALL CHAR(104,"FF0303030
3030303808080808080FFFF01010
1010101FFFF03030303030303FF"
)
260 CALL CHAR(108,"FFC0C0C0C
0C0C0C0FFFF80808080808080FFFF0
10101010101C0C0C0C0C0C0C0FF"
)
270 CALL CHAR(112,"000000181
818181800000001F1F00000000000
0FFFF000000000000F8F8181818"
)
280 CALL CHAR(116,"181818181
818181800000001F1F18181800000
0FFFF181818181818F8F8181818"
)
290 CALL CHAR(120,"1818181F1
F181818181818FFFF18181818181
8F8F8000000181818181800000"
)
300 CALL CHAR(124,"1818181F1
F0000001818181FFFF00000000000
0F8F8000000000000000000000"
)
301 DISPLAY AT(24,1):"PRESS
ANY KEY TO CONTINUE"
302 CALL KEY(0,K,S):: IF S<1
 THEN 302
303 GOSUB 800
310 FOR I=1 TO 16 :: FOR J=1
 TO 16
320 READ X :: L$(I)=L$(I)&CH
R$(X)
330 NEXT J :: NEXT I
340 RANDOMIZE :: CALL MAGNIF
Y(2)
350 DATA 96,100,100,97,96,10
0,100,97,96,100,100,97,96,10
0,100,97
360 DATA 101,96,97,102,101,9
6,97,106,105,96,97,106,105,9
6,97,102
370 DATA 101,98,99,102,101,9
8,99,110,109,98,99,110,109,9
8,99,102
380 DATA 98,107,111,99,98,10
3,103,99,98,103,103,99,98,10
7,111,99
390 DATA 96,104,108,97,96,10
0,100,97,96,100,100,97,96,10
4,108,97
400 DATA 101,96,97,102,101,9
6,97,106,105,96,97,106,105,9
6,97,102
410 DATA 101,98,99,102,101,9
8,99,110,109,98,99,110,109,9
8,99,102
420 DATA 98,107,111,99,98,10
7,111,99,98,107,111,99,98,10
7,111,99
430 DATA 96,104,108,97,96,10
4,108,97,96,104,108,97,96,10
4,108,97
440 DATA 101,96,97,106,105,9
6,97,106,105,96,97,102,101,9
6,97,102
450 DATA 101,98,99,110,109,9
8,99,110,109,98,99,102,101,9
8,99,102
460 DATA 98,107,111,99,98,10
7,111,99,98,103,103,99,98,10
```

```
3,103,99
470 DATA 96,104,108,97,96,10
4,108,97,96,100,100,97,96,10
0,100,97
480 DATA 101,96,97,106,105,9
6,97,106,105,96,97,102,101,9
6,97,102
490 DATA 101,98,99,110,109,9
8,99,110,109,98,99,102,101,9
8,99,102
500 DATA 98,103,103,99,98,10
3,103,99,98,103,103,99,98,10
3,103,99
520 FOR I=1 TO 4 :: FOR J=1
TO 16
530 READ M(I,J)
540 NEXT J :: NEXT I
550 DATA 5,6,7,8,9,10,11,12,
13,14,15,16,0,0,0,0
560 DATA 0,0,0,0,1,2,3,4,5,6
,7,8,9,10,11,12
570 DATA 0,1,2,3,0,5,6,7,0,9
,10,11,0,13,14,15
580 DATA 2,3,4,0,6,7,8,0,10,
11,12,0,14,15,16,0
590 FOR I=1 TO 16 :: READ R(
I),C(I):: NEXT I
600 DATA 25,73,25,105,25,137
,25,169,57,73,57,105,57,137,
57,169
610 DATA 89,73,89,105,89,137
,89,169,121,73,121,105,121,1
37,121,169
620 DISPLAY AT(24,1):"PRESS
ANY KEY TO CONTINUE"
630 CALL KEY(0,K,S):: IF S<1
 THEN 630
640 DISPLAY AT(1,8)ERASE ALL
:"SIXTEEN PUZZLE"
650 FOR I=1 TO 16 :: DISPLAY
 AT(I+2,7):L$(I):: NEXT I
660 DISPLAY AT(5,1)SIZE(4):"
MOVE"
670 RETURN
797 !!!!!!!!!!!!!!!!!!!!!!!!
798 ! INSTUCTIONS          !
799 !!!!!!!!!!!!!!!!!!!!!!!!
800 DISPLAY AT(1,8)ERASE ALL
:"SIXTEEN PUZZLE":TAB(8);"--
------------"
810 DISPLAY AT(4,1):"THIS PU
ZZLE IS BASED ON THE":"ORIGI
NAL INVENTED BY SAM"
820 DISPLAY AT(6,1):"LOYD OF
 AMERICA IN ABOUT":"1870. IN
 THE ORIGINAL LYOD"
830 DISPLAY AT(8,1):"OFFERED
 A PRIZE OF $1000 FOR":"A SO
LUTION. LYOD WAS SAFE AS"
840 DISPLAY AT(10,1):"HE PRE
SENTED IT SO IT COULD":"NOT
BE SOLVED! YOU ARE MUCH"
850 DISPLAY AT(12,1):"LUCKIE
R AS YOU WILL BE GIVEN":"A P
UZZLE WHICH CAN BE! BUT"
860 DISPLAY AT(14,1):"THEN,
YOU DON'T GET $1000!"
870 DISPLAY AT(16,1):"TO SOL
VE THE PUZZLE YOU MUST":"MAT
CH THE PATTERNS ON THE"
880 DISPLAY AT(18,1):"INNER
AND OUTER SQUARES. YOU":"RE-
ARRANGE INNER SQUARES BY"
890 DISPLAY AT(20,1):"USING
THE ARROW KEYS TO MOVE":"AN
ADJACENT INNER SQUARE TO"
900 DISPLAY AT(22,1):"THE BL
ANK ONE. GOOD LUCK!"
910 DISPLAY AT(24,1):"INITIA
LIZING .. PLEASE WAIT"
920 RETURN
997 !!!!!!!!!!!!!!!!!!!!!!!!
998 ! SETUP NEW PUZZLE     !
999 !!!!!!!!!!!!!!!!!!!!!!!!
1000 FOR I=1 TO 15 :: P(I)=I
 :: NEXT I :: P(16)=0
1010 FOR I=15 TO 2 STEP -1 :
: J=INT((I-1)*RND+1):: T=P(I
):: P(I)=P(J):: P(J)=T :: NE
XT I
1020 FOR I=1 TO 15 :: O(I)=P
(I):: CALL SPRITE(#P(I),P(I)
+111,2,R(I),C(I)):: NEXT I
1030 B=16 :: MOVES=0 :: BEST
=999
1040 DISPLAY AT(7,2)SIZE(3):
USING "###":MOVES :: DISPLAY
 AT(5,25)SIZE(4):" " :: DISP
LAY AT(7,25)SIZE(4):" "
1050 RETURN
1097 !!!!!!!!!!!!!!!!!!!!!!!!
1098 ! SETUP REDO PUZZLE   !
1099 !!!!!!!!!!!!!!!!!!!!!!!!
1100 FOR I=1 TO 15 :: P(I)=O
(I):: NEXT I :: P(16)=0
1110 FOR I=1 TO 15 :: CALL S
PRITE(#P(I),P(I)+111,2,R(I),
C(I)):: NEXT I
1120 B=16 :: MOVES=0
1130 DISPLAY AT(7,2)SIZE(3):
USING "###":MOVES
1140 RETURN
1197 !!!!!!!!!!!!!!!!!!!!!!!!
1198 ! MAKE MOVES           !
1199 !!!!!!!!!!!!!!!!!!!!!!!!
1200 CALL KEY(0,K,S):: IF S<
1 THEN 1200
1210 K=POS("EXDS",CHR$(K),1)
:: IF K=0 THEN 1200
1220 IF M(K,B)=0 THEN 1200
1230 S=M(K,B)
1240 CALL LOCATE(#P(S),R(B),
C(B)):: MOVES=MOVES+1
1260 P(B)=P(S):: B=S :: P(B)
=0 :: V=0
1270 FOR I=1 TO 15 :: IF P(I
)=I THEN V=V+1
1280 NEXT I :: DISPLAY AT(7,
2)SIZE(3):USING "###":MOVES
:: IF V<15 THEN 1200
1290 RETURN
1997 !!!!!!!!!!!!!!!!!!!!!!!!
1998 ! DISPLAY RESULT      !
1999 !!!!!!!!!!!!!!!!!!!!!!!!
2000 BEST=MIN(BEST,MOVES)::
DISPLAY AT(5,25):"BEST" :: D
ISPLAY AT(7,26):USING "###":
BEST
2010 DISPLAY AT(21,1):"PRESS
 'R' TO RETRY PUZZLE":"
 'N' FOR A NEW PUZZLE":"   O
R 'E' TO END"
2020 CALL KEY(0,K,S):: IF S<
1 THEN 2020
2030 K=POS("RNE",CHR$(K),1):
: IF K=0 THEN 2020
2040 ON K GOSUB 1100,1000,25
00
2050 DISPLAY AT(21,1):" ":"
":" " :: RETURN
2500 CALL CLEAR :: STOP
```

Rapid Copy

Are you tired of waiting for your disk manager to copy or format disks? If so, try Rapid Copy, the only turbo copier designed to take full advantage of the Myarc, CorComp and TI disk controller cards. Rapid Copy is lightning fast; it will copy a full double-sided double-density disk in only 1 minute and 17 seconds... and that includes formatting the target disk! On average, Rapid Copy is six times faster than the disk copy and format functions of the Myarc, CorComp, and DM1000 disk managers. Rapid Copy is simple to use and fully menu driven.

Requires: 32K, disk system, and either an Extended Basic, Editor/Assembler, or TI Writer command module.

Note: Rapid Copy is not compatible with the Myarc Geneve 9640 or Myarc Hard and Floppy Disk Controller.

Courtesy of TIsHUG

New Funnelweb Editor

Review by Geoff Trott

I have been sent a beta test version of Tony McGovern's latest update to the editor for testing with a hard disk system. The notes that came with the package are printed elsewhere in this issue, but I thought it would help if I gave some of my thoughts on this latest bit of excellent software for our very useful home computer.

Firstly I must admit that my system is not that typical as I am using an 80 column card (Mechatronics), but with luck and some effort from Garry Bowser more of you may be in the same position. Although there is a 40 column version of the editor also coming it does not have all the facilities of the 80 column version, as it lacks the features requiring the extra memory available with the 80 column cards. Although I also have a hard disk system (Myarc), this does not add as much to the system as having a RAMdisk. I happen to be basically running on a floppy disk and RAMdisk system at the moment as I need to try and recover one of my hard disks which crashed some months ago. (Oh for a bit of free time!) So I am able to test the hard disk facilities of the program but am doing all my work on floppies at the moment.

The TI-Writer package provided a first class word processing package by using the facilities of the formatter. It is true that it is a command type word processor and not a "what you see is what you get" type, but this gives a basic 40 column screen computer the ability to prepare documents to any width with relative ease. I used it for a number of years to prepare the contents of the TND, which was all in electronic form, formatted to fit our requirements, checked for errors and printed on our printer. I also happen to use Microsoft Word on a Macintosh at work so I know about all the bells and whistles that you can get in that environment, but that takes at least 1 Mbyte of memory to run while TI-Writer fits into 32 Kbytes of memory. I still enjoy using the TI99/4A and its Funnelweb environment for word processing at home.

By putting the TI-Writer editor into Funnelweb, Tony McGovern added features like the file name mail box and the ability to change case of letters with simple key strokes. This made the system so much easier to use as it became more integrated and required less typing of file names. He also improved the show directory feature along the way but the basic editor remained mainly untouched. Now he has written his own editor (based on the operation of the TI-Writer editor) which provides many more features as well as all the ones we are used to. In the process he has merged in the Euro-Writer system and expanded it to provide a configurable editor to suit almost any language with reasonable shaped characters. There is also a help facility which can provide screens of help which can also be prepared easily by the user.

Let me try and explain what is now possible and has been improved. Firstly, the editor is entered in the normal way from the menu of Funnelweb. It looks the same as the previous version at this point except for a change in the prompt line above the command line. This contains a few new entries. The new line reads:
Edit,Files,Lines,SearcH,Tabs,View,Help,STore,ReCall,Quit

Edit will return to the edit mode as before, but just pressing <enter> with nothing on the command line will return to edit mode. There are some additional ways to return to edit mode after various commands. Ctrl[1] returns to edit mode with the cursor at the point that it was when you entered command mode. This is useful with the new ability to scroll the screen while in command mode, although I am not sure how it differs from just a blank command line. It does not work after a change command. To return to where you were when you entered command mode after having completed an FS or RS, return to the edit mode and press ctrl[o]. Ctrl[2] returns to edit mode with the cursor at the top line of the current screen.

Pressing F (followed by <enter>) for Files will bring up another menu of commands. The number of these commands has also been increased and they read:
LoadF,SaveF,PrintF,LoadTemp,ShowDirectory,HardDisk,
sDPrinter,Purge
LF, SF and P work as before. Print file (PF) has some enhancements in the way of option codes which are placed in front of the device name, separated by a space. The existing option code is C for stripping out control codes. The new ones are A for appending to an existing file; M to output to a file in display fixed 128 format with MS-DOS end of line separators (<cr><lf>) and ctrl[z] end of file marker; U to output to a file in display fixed 128 format with Unix end of line separators (<lf>) and ctrl[d] end of file marker; P to send printer start up control code sequence (if these have been installed during the configuration of the editor) before sending the text; and Q to send printer reset up control code sequence (if these have been installed during the configuration of the editor) before sending the text. Load temporary file (LT) works the same as LF except that the work file name is not changed in the Funnelweb mail box. This allows other files to be merged and saves re-typing the work file name. Hard disk show directory (HD) allows a path (or disk) name to be entered (instead of just a disk number) which allows sub-directories on hard disks to be used as well as RAMdisk names such as DSKA and names of floppy disks (DSK.NAME. Note that the name must end with a period.). Once the name is entered the disk or sub-directory is catalogued as per normal. It does not give quite the same information as SD about the files but does identify emulate files and sub-directories and allows files to be marked for use and their names put in the mail boxes. Once a path is defined, it can be accessed by specifying 0 when inshow directory. Show-directory printer (DP) allows a different device from the print file (PF) printer to be used for printing the contents of the disk in show directory mode. After defining the device name, it then goes to the show directory function.

Speaking of the show directory function, this has been enhanced in several ways. There are now three files which can be marked for use. The first is the work file which is marked with <space> as before. The second is the temporary file (loaded with LT) which is marked with T. The third is the view file. Show directory now has a view file function like DiskReview (scrolling in both directions) and the viewed part of the file is stored in the VDP memory buffer and becomes marked as the B file. This is probably only an 80 column version feature. The contents of the buffer are able to be viewed inside the editor. Another nice feature is that the number of bytes unused in the editor buffer is displayed so you can see how close you are to filling the editor buffer.

Pressing L (followed by <enter>) for Lines will bring up another menu of commands. The number of commands has also been increased and they read:
Move,Copy,Delete,Show,Mark lines
The first four of these are the same as before while the last enables a line to be marked and then jumped to from anywhere in edit mode with fctn[=]. The use of these commands has been made easier by the ability to scroll the editor buffer while in command mode. This means that to find the line numbers for a move or copy of a block of lines, the lines can be displayed while constructing the command rather than remembering them. Also ctrl[m] will put the line number of the top line on the screen at the cursor position in the command line. So by using the scrolling functions and ctrl[m] it is not necessary to type in any line numbers for these commands. To show a line the same thing can be done although it is not necessary to type the S first as just a number on the command line will do it along with ctrl[2] to go to the line at the top of the screen and an empty line (or ctrl[1]) to return to where you left the edit mode.

The search functions of FS and RS have been improved by allowing the string delimiter character (/ in the old editor) to be any character other than a numeric character (0 to 9). If you wish to change or find a string of characters which include the character

"/", it is now possible to do so. Also at the end of the function there is a sound made and the editor remains in command mode. It is also possible to return to same place in the edit buffer by going to edit mode and pressing ctrl[o].

Tabs now allow up to 9 sets of tabs of which the first three are saved with the file. The particular tab set to be used is selected by number. The editor can cope with tabs from any version of TI-Writer.

View is one of the new functions which allows the contents of the view buffer to be viewed in the editor. The view keys are active for scrolling both ways. This allows viewing without going into show directory.

Help allows up to 4 screens of help to be displayed. These screens can be prepared by the user and there is a program for converting them to the correct format for the editor. As supplied by Tony, the four screens show the full character set, a quick reference for the function keys of the editor, Funnelweb entry points and Assembler language mnemonics and other useful information. The pages of help are paged with the Q and A keys.

Store and recall are very useful commands for parking the entire editor buffer into VDP memory (in less than a second) to allow some other file to be loaded and worked on. When desired the editor buffer can be recalled from the VDP memory and editing resumed. They are only available in the 80 column version because of the extra memory. It would also be nice if the contents of this buffer would remain after leaving the editor to do a format for example and was still there when the editor was re-entered. Tony might like to think about this possiblity as it could speed up preparation of documents greatly.

Quit is the last function on the main menu. This has not been changed but an additional way of leaving has been added. QQ will go directly to Funnelweb with a reminder to save the current work file first if this has not been done. This reminder is issued before a recall or purge also. A very handy warning!

If you have understood all this then you will recognise that this new editor is indeed a great improvement on the previous one in the ease of use and facilities offered, particularly in the 80 column case. I have not yet mentioned the All Characters mode which is part of this editor. When loading the editor, if you press the space bar you will receive another set of menus. The first of these allows you to choose the text editor or the program editor (eqivalent to the two editors in the Funnelweb menu). Having chosen one of these, you are then able to select to use the default 7 bit normal editor; a national 7 bit editor which just has the capability for other languages in the menus and prompts; All Characters which allows for entry of all characters with ASCII codes from 0 to 254; and TI Euro-Writer for files from Europe. For normal use the first option is all that is required. If you want access to IBM graphics for example, you can choose the All Characters mode. The help files tell you what key combinations to press to get all the non-standard characters. There are some special key presses for putting accents on vowels which are listed in the help files but there is one key press which applies to all and which I should mention. You can freeze the bottom of the screen below the line on which the cursor is placed with ctrl[f]. This allows some text to be kept on the screen for a reminder while you type in anywhere else in the document.

Well I see that I have only 11,780 bytes left in the buffer so I will conclude by saying that I think that this editor will provide serious users with much joy and save them many frustrations and much time. I would like to thank Tony McGovern for his efforts on our behalf and look forward to the release of version 5 of Funnelweb. O

TI-Writer Font Maker

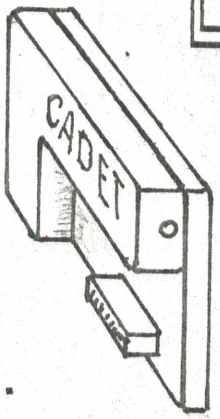
by James Stringfellow, Paris France

```
!************************
!*                      *
!* TIWRITER FONT MAKER  *
!*                      *
!*   VERSION 1 1987     *
!*                      *
!* JAMES STRINGFELLOW   *
!*                      *
!************************
!
CALL CLEAR :: CALL SCREEN(5):: CALL CHAR(128,"FF818
181818181FF",122,RPT$("0",16),129,RPT$("F",16))
FOR I=5 TO 12 :: CALL COLOR(I,2,11):: NEXT I :: CALL
COLOR(13,2,11,0,5,5,1,1,13,3,2,13,4,2,13):: P$=RPT$(CH
R$(128),16)
X=64 :: CH=128 :: FOR R=4 TO 17 :: DISPLAY AT(R,4):
STR$(X);TAB(7);P$ :: X=X/2 :: IF X<1 THEN X=64
NEXT R :: CALL VCHAR(1,31,28,96):: Z$=RPT$(CHR$(32),
28)
DISPLAY AT(21,1):"zMovezwithzzWzEzRzSzDzZzXzCz":"zQ
zTogglezonzoffzzz@zstopzzz":"zKzclearzzPzprintzzMzmi
rrorz":"zzzzzzzzzzzzzzzzzzzzzzzzzzz" :: M$,M2$=""
DISPLAY AT(1,7):"zTIWRITERzFONTSz":Z$:Z$ :: DISPLAY
AT(18,1):Z$:Z$:Z$ :: R=4 :: C=9 :: K=75 :: CALL SPRITE(
#1,128,16,25,65)
REM
CALL KEY(3,K,S):: Y=(K=69 OR K=82 OR K=87)-(K=88 OR
K=90 OR K=67):: X=(K=83 OR K=87 OR K=90)-(K=68 OR K=82
OR K=67) :: IF K=81 THEN 190 :: IF K=80 THEN 200 : IF K=
75 THEN 120
IF K=64 THEN 370 :: IF K=77 THEN 330 :: R=R+Y :: C=
C+X :: R=R-(R<4)+(R>17):: C=C-(C<9)+(C>24):: CALL HCHAR
(R, C,CH):: CALL LOCATE(#1,R*8-7,C*8-7):: GOTO 170
CH=CH+1+(CH=129)*2 :: CALL HCHAR(R,C,CH):: FOR I=1
TO 30 :: NEXT I :: GOTO 70
GOSUB 350 :: FOR C=9 TO 24 :: X=64 :: FOR R=4 TO 10
:: CALL GCHAR(R,C,G):: FG=129 THEN A=A+X
X=X/2 :: NEXT R :: FOR J=1 TO LEN(STR$(A)):: CALL
VCHAR(J,C,ASC(SEG$(STR$(A),J,1))):: NEXT J :: M2$=M2
$&CHR$(A):: A=0 :: NEXT C
FOR C=9 TO 24 :: X=64 :: FOR R=11 TO 17 :: CALL GCH
AR(R,C,G):: IF G=129 THEN A=A+X
X=X/2 :: NEXT R :: FOR J=1 TO LEN(STR$(A)):: CALL
VCHAR(17+J,C,ASC(SEG$(STR$(A),J,1))):: NEXT J :: M$=
M$&CHR$(A):: A=0 :: NEXT C :: CALL DELSPRITE(ALL):: CALL
MAGNIFY(1)
CALL SPRITE(#1,128,16,185,233):: DISPLAY AT(24,6):"
zzzzzzzDoublezDensityzN" :: ACCEPT AT(24,28)VALIDATE
("YN")SIZE(-1):Q$ :: K=75 :: IF Q$="Y" THEN K=K+1
OPEN #1:"RS232.BA=4800.DA=8.PA=N"
PRINT #1:CHR$(27)&"A"&CHR$(7)
PRINT #1:CHR$(27)&CHR$(K)&CHR$(16)&CHR$(0)&M2$&CHR$
(13)
PRINT #1:CHR$(27)&CHR$(K)&CHR$(16)&CHR$(0)&M$ ::
CLOSE #1
DISPLAY AT(24,6)SIZE(23):"Printzagainzyeszorznozn"
:: ACCEPT AT(24,28)VALIDATE("YN")SIZE(-1):Q$ ::
IF Q$="Y" THEN 235
DISPLAY AT(24,6):"zzzzzzzzzzSavezyesznozN" :: ACCEPT
AT(24,28)VALIDATE("YN")SIZE(-1):Q$ :: IF Q$="N" THEN
140
CALL LOCATE(#1,185,145):: DISPLAY AT(24,2):"zzzFile
namezDSK" :: ACCEPT AT(24,17)SIZE(-12):F$ :: IF F$=""
THEN 140 :: OPEN #2:"DSK"&F$,FIXED 80
PRINT #2:CHR$(27)&"K"&CHR$(16)&CHR$(0)&M2$
PRINT #2:CHR$(27)&"K"&CHR$(16)&CHR$(0)&M$ :: CLOSE
#2 :: GOTO 140
FOR R=4 TO 17 :: X=64 :: FOR C=9 TO 24 :: CALL GCHAR
(R,C,CH):: IF CH=128 THEN CALL HCHAR(R,C,129,1)ELSE
IF CH=129 THEN CALL HCHAR(R,C,128,1)
NEXT C :: NEXT R :: CH=CH+1+(CH=129)*2 :: GOTO 140
DATA 80,76,69,65,83,69,32;87,65,73,84
CALL DELSPRITE(#1):: CALL MAGNIFY(2):: FOR I=2 TO 22
STEP 2 :: READ Y :: CALL SPRITE(#I,Y,16,I*8-7,I+I*8-7
+16):: NEXT I :: RESTORE :: RETURN
END
```

O

CADET

CONSOLE EXPANSION



Plugs into I/O port on right of the console.
Compatable with all software from tapes and modules.
Features:-

PRINTER PORT. Drive a printer directly from Basics or print text files.
SOFTWARE IN ROM:
Word Processor like the original TI-Writer. Type and print letters, assignments etc.
Text Formatter like original TI version. Add "class" to your printed text.
Tape Games Loader to load and run specially taped assembly games.
32K MEMORY EXPANSION for use by the above software.
ADDITIONAL 32K RAM divided into two devices to store text for later reuse. Supercap backup maintains power to memory for weeks.
I/O EXTENSION. Attach a speech synthesizer or ramdisk. Ramdisk requires optional adaptor card.

Limited number available to order at $100 each.
Ramdisk adaptor card $20.
Contact Col Christensen on (07)2847783.

SOUND F/X

Listen to your computer sing, talk, scream and even play music tracks!
With Sound F/X you can play previously recorded audio files on a standard TI-99/4a or Geneve
<u>without any additional hardware!</u> The most incredible software experience in TI history!

The most incredible thing happened in our labs... we discovered how to make a standard TI-99/4a and Geneve 9640 play true digital sound through your existing monitor, without requiring you to purchase any new hardware. We call our discovery Sound F/X. You'll call it "amazing". But we didn't stop there... taking our incredible breakthrough one step further we developed a vital link that would connect Sound F/X to the outside world. This development is a sophisticated utility that converts industry standard sound files originally recorded using IBM, Macintosh and Amiga personal computers into Sound F/X files. And it's included free with the Sound F/X program. Other features include a convenient disk cataloging utility, a variable playback speed control and a memory buffer report (varies with your installed equipment). ***Only $14.95***

Sound F/X is compatible with the TI-99/4a, Geneve 9640 and Myarc HFDC. The program automatically takes advantage of memory additinal found in the Geneve 9640, the Memex card, 80 column devices, and the Supercart and Superspace modules. Operating requirements: TI-99/4a with 32K, disk system, and either an Extended Basic, Editor/Assembler or TI Writer module -- or -- Geneve 9640 with Extended Basic, Editor/Assember modules or Exec. Included with Sound F/X is a small sampling of Sound Bytes (recorded files for playback).

<u>Sound F/X Sound Bytes</u>

There are thousands of Sound F/X available world-wide, and they can typically be obtained through user groups and online services such as CompuServe, Genie and local bulletin boards. If you prefer, you can purchase our own two disk Sound F/X Sound Bytes for only $3.95 each.

Sound Bytes #1: Cartoons. This animated collection features Bart Simpson plus Tweety Bird, Porky Pig, Yosemite Sam and more.
Sound Bytes #2: Science Fiction. Includes sound bytes from *Star Trek, Star Wars, 2001: A Space Odyssey*, plus more hi-tech stuff.
Sound Bytes #3: Miscellaneous. Includes snippets such as Tarzan yodeling and Robin Williams screaming "Good Morning Vietnam".

<u>Super Special:</u> Order Sound F/X with all three Sound Byte Collections for only $21.95,
or all three Sound Byte Collections alone for just $9.95.

TEXAMENTS
53 Center Street, Patchogue, New York 11772

BUG BYTES

TI BRISBANE USER GROUP
P.O. BOX 3051
CLONTARF M.D.C.
QLD AUST 4019.

43
GRAEME MORRIS
P.O. BOX 371
CLEVELAND QLD 4163